Lib.

Proceedings

TENTH ANNUAL ALLERTON CONFERENCE ON CIRCUIT AND SYSTEM THEORY

October 4-6, 1972

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

Allerton Conference on Circuit and
System Theory,
10th Annual, 1972

[redacted]

~~DEPARTMENT OF~~
~~COMPUTER SCIENCE~~
LIBRARY

PROCEEDINGS

TENTH ANNUAL ALLERTON CONFERENCE
ON CIRCUIT AND SYSTEM THEORY

J. B. Cruz, Jr.
C. L. Chen

Conference Co-Chairman

Conference held

October 4, 5, 6, 1972
Allerton House
Monticello, Illinois

Sponsored by the

DEPARTMENT OF ELECTRICAL ENGINEERING
and the
COORDINATED SCIENCE LABORATORY
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

FOREWORD

The Tenth Annual Allerton Conference on Circuit and System Theory was held at the Allerton House on October 4-6, 1972. The Conference was co-sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the Conference.

In addition to the large number of submitted papers, there were several organized sessions of invited papers. These sessions were: Artificial Intelligence, organized by K. S. Fu; Large Scale Networks, organized by I. T. Frisch; Coding Theory, organized by J. K. Wolf; Development in Computer Architecture, organized by C. V. Ramamoorthy; Parameter Identification in Distributed Processes, organized by R. E. Klein; and Linear Integrated Circuits and Their Applications, organized by T. N. Trick. We are grateful to the organizers for their time and effort on behalf of the Conference. We thank the authors in these sessions for their prompt response and their excellent contributions. We are grateful to the session chairmen and all the authors for their effort and contribution to the success of the Allerton Conference.

Dr. James H. Mulligan, Jr. delivered a very stimulating address entitled "Research, Engineering, and National Goals." Professor B. R. Myers introduced the keynote speaker for the tenth year. We are much indebted to them. We thank Dr. G. David Forney, Jr. for his talk at the joint meeting with the IEEE Group on Information Theory.

The support and encouragement of Professor E. C. Jordan, Head of the Department of Electrical Engineering, and Professor C. L. Coates, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We thank our colleagues for their generous assistance in organizing the Conference: R. T. Chien, A. H. Haddad, P. V. Kokotovic, W. Mayeda, G. Metze, M. Mintz, W. R. Perkins, F. P. Preparata, T. N. Trick and N. Wax.

We are grateful to Mr. Harold Lawler for his handling of the business aspects of the Conference. We thank Rose Harris, Sherry Kallenbach, and Rosa Townsend for their help in the preparations of the Conference. Thanks are also due Mr. W. T. Warren for coordinating the projectionists and the visual aid equipment. Special thanks are due to Mrs. Dee Keel for her expert and skillful coordination of the Conference arrangements.

J. B. Cruz, Jr.
C. L. Chen

TABLE OF CONTENTS

Foreword iii

I-A: ARTIFICIAL INTELLIGENCE
Chairman and Organizer: K. S. Fu, Purdue University

ARTIFICIAL INTELLIGENCE -- PROMISES AND PROBLEMS 1
L. D. Harmon, Bell Telephone Laboratories, Murray Hill, New Jersey

THE LINGUISTIC APPROACH TO PATTERN RECOGNITION AS A BASIS FOR ADAPTIVE PROGRAM BEHAVIOR 2
R. H. Anderson, USC Information Sciences Institute, Marina del Rey, California

ON THE AUTOMATIC GENERATION OF HYPOTHESIS 11
R. C. T. Lee and C. L. Chang, National Institutes of Health, Bethesda, Maryland

THE DECOMPOSITION PRINCIPLE FOR THEOREM PROVING SYSTEMS . . 20
C. L. Chang, National Institutes of Health, Bethesda, Maryland

I-B: LARGE SCALE SYSTEMS
Chairman: E. J. Davison, University of Toronto, Canada

STABILITY REGIONS OF LARGE-SCALE SYSTEMS 29
Stein Weissenberger, University of Santa Clara

PARTIAL IDENTIFICATION OF LARGE-SCALE SYSTEMS 39
R. P. Guidorzi and G. Marro, University of Bologna, Italy

A FUNCTIONAL APPROACH TO LARGE-SCALE DYNAMICAL SYSTEMS . . 48
M. N. Ransom and R. Saeks, University of Notre Dame

SUBOPTIMAL CONTROL OF COUPLED TIME DELAY SYSTEMS 56
M. Jamshidi, Pahlavi University, Iran

I-C: APPLICATION OF ESTIMATION AND FILTERING
Chairman: D. L. Snyder, Washington University

OPTIMUM SAMPLE RATES FOR DATA ESTIMATION 66
L. C. Ludeman, New Mexico State University

ON THE DATA STORAGE PROBLEM IN SEQUENTIAL DISCRETE-TIME SMOOTHING 68
J. S. Meditch, University of California, Irvine

INTERMEDIATE RANGE PEAK LOAD DEMAND FORECASTING 76
K. R. Shah, B. V. Baliga and Y. Yavin, Virginia Polytechnic Institute and State University

PERFORMANCE COMPARISON OF SUBOPTIMAL KALMAN FILTERS MODELED FOR A CONTINUOUS NONLINEAR SYSTEM 85
H. L. Pastrick, US Army Missile Command, Redstone Arsenal, Alabama

SEQUENTIAL PARAMETER ESTIMATION USING PSUEDO-INVERSE . . . 95
R. P. Nayak and E. C. Foudriat, Marquette University

ON THE ESTIMATION AND DETECTION FOR OPTICAL TURBULENT CHANNELS 97
R. O. Harger and K. Kurimoto, University of Maryland

ON THE ESTIMATION PROBLEM AS VIEWED FROM INFORMATION THEORY 99
P. Kalata, Illinois Institute of Technology and R. Priemer, University of Illinois at Chicago Circle

I-D: FAULT DIAGNOSIS
Chairman: J. Nievergelt, University of Illinois at Urbana

A MULTIPLE FAULT DETECTION ALGORITHM FOR COMBINATIONAL LOGIC CIRCUITS 101
S. S. Yau and S. C. Yang, Northwestern University

DETECTION OF INTERMITTENT FAULTS IN COMBINATIONAL CIRCUITS 111
S. Kamal, Wayne State University and C. V. Page, Michigan State University

COMPLETE TEST SETS FOR LOGIC FUNCTIONS 121
S. M. Reddy, University of Iowa

SOME RESULTS ON TOTALLY SELF-CHECKING CHECKERS 129
S. M. Reddy, University of Iowa

A THEORY AND A PROCEDURE FOR THE DETECTION OF MULTIPLE FAULTS IN COMBINATIONAL CIRCUITS 133
S. R. Vishnubhotla and Y. H. Chuang, Washington University

II-A: LARGE SCALE NETWORKS
Chairman and Organizer: I. T. Frisch, Network Analysis Corporation

NETWORKS AND VEHICLE ROUTING FOR MUNICIPAL WASTE COLLECTION 143
E. J. Beltrami and L. D. Bodin, State University of New York

GRAPH THEORY PROBLEMS IN PHYSICS 152
D. J. Kleitman, Massachusetts Institute of Technology

GRAPH THEORY IN DIAGNOSTIC PROBLEMS 159
W. Mayeda, University of Illinois at Urbana

RELIABILITY OF NETWORKS 165
R. Van Slyke, Network Analysis Corporation, Glen Cove, New York

II-B: CODING THEORY
Chairman and Organizer: J. K. Wolf, Polytechnic Institute of Brooklyn

GENERALIZED FINITE-GEOMETRY CODES 166
C. R. P. Hartmann and L. D. Rudolph, Syracuse University

CODING THEORY AND THE COMPUTER 175
R. T. Chien, University of Illinois at Urbana

COMPUTATION BOUNDS ON 3-GROUP BOOTSTRAP DECODING 178
D. J. Costello, Jr., Illinois Institute of Technology and F. Jelinek, Cornell University

ADAPTIVE HYBRID SEQUENTIAL DECODING 185
L. Bahl, J. Cocke, F. Jelinek and J. Raviv, IBM Research Center, Yorktown Heights, New York

STRUCTURAL ANALYSIS OF CONVOLUTIONAL CODES VIA DUAL CODES 186
G. David Forney, Jr., Stanford Electronics Labs.

II-C: LINEAR SYSTEMS
Chairman: W. S. Levine, University of Maryland

PROPERTIES OF LINEAR TIME-INVARIANT MULTIVARIABLE SYSTEMS SUBJECT TO ARBITRARY OUTPUT AND STATE FEEDBACK 196
E. J. Davison and S. H. Wang, University of Toronto, Canada

L^2-STABILITY OF DISTRIBUTED FEEDBACK SYSTEMS: SINGULAR PERTURBATION 205
J. F. Barman, F. M. Callier and C. A. Desoer, University of California, Berkeley

THE ANALYTIC EVALUATION OF e^{At} USING A COMPUTER 209
W. C. Miller and M. Sid-Ahmed, University of Windsor, Canada

SMITH-MCMILLAN FORM FROM STATE-SPACE REPRESENTATION FOR NON-MINIMAL SYSTEMS 211
O. M. Grasselli, Centro di Studio del Sistemi di Controllo e Calcolo Automatici, Italy

ON THE DETERMINATION OF THE CAUSAL PART OF A NONCAUSAL SYSTEM 221
R. M. DeSantis, Ecole Polytechnique, Montreal, Canada

ON THE OPTIMAL FEEDBACK CONTROLLERS FOR SYSTEMS ON NEUTRAL TYPE 231
S. G. Chow and H. N. Koivo, University of Toronto, Canada

MODELING A SET OF SEQUENTIAL DATA 238
J. G. Lin, Columbia University

II-D: PATTERN RECOGNITION AND INFORMATION RETRIEVAL
Chairman: S. S. Yau, Northwestern University

ON GRAPH THEORY AND CLUSTER ANALYSIS 248
Z. Chen and K. S. Fu, Purdue University

DESIGN OF THRESHOLD ELEMENTS USING A POLYNOMIAL-THRESHOLD APPROACH 257
M. H. Loew and K. S. Fu, Purdue University

A MODE ESTIMATING PROCEDURE AND AN OPTIMUM RECLUSTERING ALGORITHM 269
D. A. Michalopoulos, California State University, Fullerton

ON THE OPTIMAL PROPERTIES OF A PARTITION ALGORITHM 271
P. C. Yue and C. K. Wong, IBM Thomas J. Watson Research Center, Yorktown Heights, New York

PARALLEL GENERATION OF BINARY SEARCH TREES 277
C. K. Wong and S. K. Chang, IMB Thomas J. Watson Research Center, Yorktown Heights, New York

A MARKOV ALGORITHM COMPUTER: AN EDUCATIONAL TOOL FOR COMPUTABILITY 282
M. E. Kaliski and N. R. Kashdan, The City College of New York

A TELEGRAPHICS SYSTEM FOR THE ARDS-100A 283
P. J. Caudill, R. G. Nestor, C. A. Harlow and S. J. Dwyer, III, University of Missouri-Columbia

III-A: PROCESSING OF STOCHASTIC SIGNALS
Chairman: J. S. Meditch, University of California, Irvine

ON THE REPRESENTATION OF BIVARIATE DISTRIBUTIONS 293
H. Derin and J. B. Thomas, Princeton University

REDUCED MEMORY INFORMATION PROCESSORS FOR POINT PROCESSES 303
I. Rubin, University of California, Los Angeles

GEOMETRY OF QUANTUM SIGNAL DETECTION 313
R. O. Harger and M. C. Kim, University of Maryland

APPROXIMATELY OPTIMUM DETECTION OF LINEAR PROCESSES 315
L. F. Eastwood, Jr. and R. Lugannani, Princeton University

THE DIGITAL RECONSTRUCTION OF MULTIDIMENSIONAL SIGNALS FROM THEIR PROJECTIONS 326
R. M. Mersereau, Massachusetts Institute of Technology

THE PRELIMINARY DESIGN OF A MULTICHANNEL COGNITIVE SIGNAL PROCESSOR AND SIMULATOR 335
J. M. Banda, University of Minnesota

III-B: ECONOMIC AND ECOLOGICAL MODELING
Chairman: W. R. Perkins, University of Illinois at Urbana

A DYNAMIC LABOR ALLOCATION MODEL INCLUDING EFFICIENCY . . . 344
M. H. Mickle and W. G. Vogt, University of Pittsburgh

A STOCHASTIC MODEL FOR A DYNAMIC LEONTIEF SYSTEM 351
W. G. Vogt and M. H. Mickle, University of Pittsburgh

APPLICATION OF DISCRETE-DATA SYSTEM TECHNIQUES IN A FINANCIAL PROBLEM OF INSURANCE 357
J. W. Caron, T. S. Roy and D. Tabak, The Travelers Insurance Co., Hartford, Connecticut

MATHEMATICAL MODELING ON AN AQUATIC ECOSYSTEM I 367
N. N. Gupta, Wright State University and James Houdeshell, General Motors Corporation, Dayton, Ohio

A SIMULATION OF BIOLOGICAL EVOLUTION BY INNOVATIONS IN AN ARTIFICIAL LEARNING SYSTEM 374
S. J. Wan, University of California, Irvine and L. A. Wan, Sycom, Inc., Anaheim, California

III-C: COMPUTER-AIDED DESIGN
Chairman: M. E. Van Valkenburg, Princeton University

A COMPUTER-GENERATED SEMICONDUCTOR MODEL 383
L. C. McAfee, Jr., The University of Michigan

TIME DOMAIN ANALYSIS OF GENERAL PARAMETER FILTERS USING NUMERICAL INVERSION OF THE LAPLACE TRANSFORM 392
M. E. Mokari-Bolhassan, Pahlavi University, Iran

THE SIMULATION OF CONTINUOUS TRANSFER FUNCTIONS BY USE OF PCAP 403
S. M. Scarborough, Mississippi State University

MIXED DOMAIN TRANSIENT ANALYSIS OF LARGE NON-LINEAR NETWORKS 404
C. B. Frye, Jr. and M. J. Apfelbaum, GTE Sylvania Inc., Needham Heights, Massachusetts

NONLINEAR OPTIMIZATION FOR INTEGER PROBLEMS 414
J. Opaćič, Bell Telephone Laboratories, Whippany, New Jersey and D. Tabak, RPI of Connecticut, Hartford, Connecticut

III-D: CODING APPLICATIONS

Chairman: R. T. Chien, University of Illinois at Urbana

FURTHER RESULTS ON A LARGE CLASS OF NON-BCH CYCLIC CODES FOR MULTIPLE RANDOM-ERROR CORRECTION 416
H. T. Hsu, The Pennsylvania State University

A NOTE ON SYSTEMATIC NON-SEPARATE ARITHMETIC CODES 426
M. R. Varanasi and T. R. N. Rao, University of Maryland

CHANNEL MODELING AND BURST CHARACTERISTICS 434
S. Tsai, University of Illinois at Chicago Circle

NOTE ON THE BEST MATCH PROBLEM 443
L. D. Rudolph, K. G. Mehrotra and H. Schwarzlander, Syracuse University

III-E: AUTOMATA

Chairman: D. Schertz, Bradley University

UNIVERSAL TESSELLATION AUTOMATA: AN APPLICATION OF COMBINATORY LOGIC TO AUTOMATA THEORY 452
W. I. Grosky, Georgia Institute of Technology

A PROCEDURE FOR DESCRIBING DISCRETE PROCESSES 462
M. R. Schaffner, Massachusetts Institute of Technology

EXPERIMENTS ON SEQUENTIAL MACHINES 471
G. F. Beckhoff, The University of Western Ontario, Canada

ANALYTICAL REPRESENTATION OF FORMAL LANGUAGES 481
W. J. Kelly and R. J. Leake, University of Notre Dame

IV-A: CONTROL AND OPTIMIZATION

Chairman: J. E. Prussing, University of Illinois at Urbana

ON THE CONTROL OF LINEAR SYSTEMS USING TWO LEVEL PERIODIC OUTPUT FEEDBACK 486
R. V. Welch, NASA Goddard Space Flight Center, Greenbelt, Maryland and W. S. Levine, University of Maryland

ADAPTIVE CONTROL AND IDENTIFICATION VIA LIAPUNOV'S DIRECT METHOD 496
R. V. Monopoli, University of Massachusetts

DUAL VARIATIONAL TECHNIQUES AND THE EPSILON METHOD 498
W. L. Chan, M. D. Lodaya and D. J. Ewing, Jr., The University of Toledo

AIRCRAFT PARAMETER IDENTIFICATION: A VARIATIONAL PROCEDURE 507
R. E. Bach, Jr., Northeastern University, Boston, Massachusetts, Susan E. Post, U.S. Army AMRDL, Moffett Field, California and Lt. J. E. Tucker, Nval Postgraduate School, Monterey, California

ON-LINE SYSTEM IDENTIFICATION 509
N. K. Sinha, McMaster University, Hamilton, Canada

APPLICATION OF FOUR METHODS FOR APPROXIMATING OPTIMAL FEEDBACK GAINS 520
J. H. Burghart, State University of New York at Buffalo

ON A LINEAR DISCRETE RENDEZVOUS PROBLEM 530
D. H. Chyung, University of Iowa

ON OPTIMAL FEEDBACK CONTROL OF STOCHASTIC SYSTEMS AND CERTAIN ASSOCIATED MATHEMATICAL PROBLEMS 532
N. U. Ahmed, Ottawa University, Canada

IV-B: ROUTING AND SCHEDULING
Chairman: S. Muroga, University of Illinois at Urbana

LOCAL BALANCE IN QUEUEING NETWORKS WITH INTERDEPENDENT SERVERS 534
M. Chandy, The University of Texas at Austin

A COMPUTER-AIDED ROUTING TECHNIQUE FOR APPLICATION IN MANUFACTURING 543
W. J. Rychlewski and R. W. McLaren, University of Missouri-Columbia

ANALYSIS OF A SIMPLE SCHEDULING ALGORITHM FOR A MULTI-PROCESSING COMPUTER SYSTEM 545
K. L. Krause and V. Y. Shen, Purdue University

SCHEDULING IN A COMPUTER SYSTEM WITH LIMITED MEMORY . . . 554
L. S. Cheung, Purdue University and Y. E. Chen, Bell Telephone Laboratories, Columbus, Ohio

DISJOINT REPRESENTATION OF TREE REALIZABLE SEQUENCES . . . 562
S. Kundu, IBM Thomas J. Watson Research Center, Yorktown Heights, New York

IV-C: COMMUNICATION SYSTEMS
Chairman: J. B. Thomas, Princeton University

UPPER AND LOWER BOUNDS ON THE PROBABILITY OF ERROR IN DIGITAL SYSTEMS 568
L. E. Jankauskas and R. Lugannani, Princeton University

FINITE-PULSE STABILITY OF INTERCONNECTED SYSTEMS WITH COMPLETE-RESET PULSE FREQUENCY MODULATORS 579
H. Ö. Gülçür and A. U. Meyer, Newark College of Engineering

THE ADVANTAGES OF COMBINING PHASE AND INJECTION LOCK . . . 587
M. Sandler, The Cooper Union, New York

TRANSIENT CHARACTERISTICS OF A HYBRID PHASE-LOCKED-LOOP . . 596
T. L. Stewart and W. M. Hammond, Bradley University

AUTOMATIC EQUALIZATION FOR QUADRATURE AMPLITUDE MODULATION SYSTEMS 598
Y. Sato, Nippon Electric Co., Ltd., Japan

IV-D: LOGIC DESIGN AND SWITCHING THEORY
Chairman: L. D. Rudolph, Syracuse University

A CELLULAR REALIZATION OF FINITE STATE MACHINES USING PARTITIONS 608
S. F. Averitt and J. W. Gault, North Carolina State University

CELLULAR ARRAYS FOR PERFORMING CONVOLUTIONS 616
R. DeMori, Istituto di Elettrotecnica Generale, Italy

THE RELATION OF EQUIVALENCE FOR POST ALGEBRAS 625
E. G. DuCasse and G. A. Metze, University of Illinois at Urbana

PRELIMINARY REPORT ON CONNECTIONS OF POST FUNCTIONS WITH SETTLEMENT OPERATORS 635
E. G. DuCasse and G. A. Metze, University of Illinois at Urbana

SYNTHESIS OF ASYNCHRONOUS CIRCUITS USING TRANSITION SENSITIVE FLIP-FLOPS 637
J. G. Bredeson and P. T. Hulina, Pennsylvania State University

SYNTHESIS OF ASYNCHRONOUS MACHINES USING FLIP-FLOPS AND CONTROLLED EXCITATION 647
Y. H. Chuang and S. Das, Washington University

ON COMPLETELY AND INCOMPLETELY SPECIFIED DIAGNOSABLE MACHINES 649
S. C. Lee, University of Houston

OPTIMAL SYSTEM DESIGN USING DIODE ARRAYS AS LOGIC STRUCTURES AND READ-ONLY MEMORIES 650
A. D. Sypherd, North American Rockwell Microelectronics Company, Anaheim, California

IV-E: NETWORK THEORY
Chairman: C. A. Desoer, University of California at Berkeley

AN ABSOLUTE BOUND ON LIMIT CYCLES DUE TO ROUND-OFF ERRORS IN DIGITAL FILTERS 659
J. L. Long and T. N. Trick, University of Illinois at Urbana

DISTRIBUTED R-C TIMING CIRCUITS IN MULTIVIBRATORS 667
R. A. Hayes and J. J. Bourquin, University of Missouri-Rolla

CONSTANT DELAY NETWORKS - AN INTEGRAL APPROACH 676
K. S. Bajaj, Detroit Institute of Technology

ACTIVE LC-LESS BANDPASS FILTER USING TRANSISTOR LOWPASS CHARACTERISTIC 685
G. A. Paphitis, National Broadcasting Company, Chicago and T. Murata, University of Illinois at Chicago Circle

SYNTHETIS OF TWO-ELEMENT-KIND CASCADE NETWORKS 692
T. N. Lee, The George Washington University and D. P. Brown, The University of Wisconsin

ON THE ANALYSIS OF SCATTERING NETWORKS 700
J. Shekel, Jerrold Electronics Corporation, Hatboro, Pennsylvania

ON A DIGITAL OSCILLATOR 709
D. Mitra and H. D. Montgomery, Bell Laboratories, Murray Hill, New Jersey

KEYNOTE ADDRESS

Introduction: B. R. Myers, University of Notre Dame

Speaker: James H. Mulligan, Jr.
Secretary and Executive Officer
National Academy of Engineering
Washington, D. C.

V-A: DEVELOPMENT IN COMPUTER ARCHITECTURE
Chairman: R. E. Merwin, U.S. Army
Organizer: C. V. Ramamoorthy, University of California at Berkeley

THE IMPACT OF THEORY ON SYSTEMS DESIGN 724
P. J. Denning, Purdue University

PERSPECTIVE AND STATUS OF LARGE SCALE DATA PROCESSING SYSTEMS 733
R. E. Merwin, Safeguard System Office, Washington, D. C.

MINICOMPUTER SYSTEMS DESIGN FOR REAL-TIME PATIENT MONITORING 742
S. S. Yau, G. F. DePalma and G. W. Dietrich, Northwestern University

HARDWARE MAINTAINABILITY AND SOFTWARE RELIABILITY OF ELECTRONIC SWITCHING SYSTEMS 752
H. Y. Chang, Bell Laboratories, Naperville, Illinois

TOWARDS A THEORY OF PIPELINED COMPUTING SYSTEMS 759
C. V. Ramamoorthy and S. S. Reddi, University of California at Berkeley

V-B: PARAMETER IDENTIFICATION IN DISTRIBUTED PROCESSES
Chairman and Organizer: R. E. Klein, University of Illinois at Urbana

AN APPLICATION OF IMPLICIT FUNCTION THEORY FOR PARAMETER IDENTIFICATION 769
J. R. Cannon, University of Texas at Austin

A DISTRIBUTED REPRESENTATION OF UPPER TORSO SAGITAL PLANE BENDING RESPONSE: MODELING AND PARAMETER IDENTIFICATION 774
L. D. Metz, University of Illinois at Urbana

HYBRID IMPLEMENTATION OF PARTIAL DIFFERENTIAL EQUATIONS FOR PARAMETER IDENTIFICATION AND MODEL MATCHING 783
M. J. Wozny, Oakland University

A TECHNIQUE FOR PARAMETER IDENTIFICATION IN A CLASS OF HYPERBOLIC DISTRIBUTED PROCESSES 784
R. E. Klein, University of Illinois at Urbana

V-C: LINEAR INTEGRATED CIRCUITS AND THEIR APPLICATIONS
Chairman and Organizer: T. N. Trick, University of Illinois at Urbana

IMPLANT ELECTRONICS FOR MEDICAL APPLICATIONS 791
W. H. Ko, Case Western Reserve University

INTEGRATED CIRCUITS FOR BIO-MEDICAL APPLICATIONS 792
J. Berger, Stanford University

NEW DEVELOPMENTS IN MONOLITHIC VOLTAGE REGULATORS 802
C. T. Nelson, National Semiconductor Corporation, Santa Clara, California

ANALOG AND DIGITAL FILTERING IN MULTIPLEX COMMUNICATION SYSTEMS 810
C. F. Kurth, Bell Telephone Laboratories, North Andover, Massachusetts

V-D: GAMES AND SYSTEMS WITH VECTOR-VALUED CRITERIA
Chairman: Max Mintz, University of Illinois at Urbana

A CONTRADICTORY PROPERTY OF THE NASH SOLUTION FOR TWO STOCHASTIC NONZERO-SUM GAMES 819
T. Basar, Harvard University

HIERARCHICAL CONTROL OF OPTIMAL SUBSYSTEMS 828
P. D. Shea, J. S. Shafran and M. A. Radke, TRW Systems, Redondo Beach, California

ON A CLASS OF LINEAR THREE-PERSON COOPERATIVE DIFFERENTIAL GAMES -- POLLUTION CONTROL 831
J. G. Lin, Columbia University

OPTIMALITY FOR A MULTICRITERION PERTURBED SYSTEM 834
A. Haurie and J. L. Goffin, Ecole des Hautes Etudes Commerciales, Montreal, Canada

OPTIMAL CONTROL WITH INCOMPLETE STATE FEEDBACK 836
C. F. Martin, Ames Research Center, NASA, Moffett Field, California

ARTIFICIAL INTELLIGENCE -- PROMISES AND PROBLEMS

LEON D. HARMON
Bell Telephone Laboratories
Murray Hill, New Jersey

ABSTRACT

Man's machines are behaving more and more like him in certain respects. In some cases, like game playing and theorem proving, artificial intelligence can at times outstrip the human. But differences between brains and computers are much more apparent than are similarities, both in behavior and in mechanism. These distinctions and some of the problems of understanding nervous systems and of designing "intelligent" automata will be discussed, together with a survey of the principal branches of machine intelligence as they have been developed to date.

THE LINGUISTIC APPROACH TO PATTERN RECOGNITION AS A BASIS FOR ADAPTIVE PROGRAM BEHAVIOR

ROBERT H. ANDERSON
USC Information Sciences Institute
Marina del Rey, California

ABSTRACT

A brief survey is given of the use of generalized linguistic techniques in performing pattern recognition. Programs whose behavior is entirely syntax-directed are quite flexible; their behavior is altered by modifying the set of syntax rules. Adaptive behavior is achieved by automatic blending of new rules into the existing set of rules.

INTRODUCTION

There is a growing body of literature on the use of linguistic techniques in pattern recognition. Recent reviews of this literature are contained in Miller and Shaw [1], Evans [2], Fu and Swain [3], and Uhr [4]. The purpose of this paper is to stress two aspects of linguistic pattern recognition which are particularly relevant to artificial intelligence:

1. *Broad applicability;* the use of sets of pattern-replacement rules in analyzing patterns is not limited to "traditional" spatial or time-sequence patterns. They are also applicable to analyzing and operating upon patterns of data and interrelationships occurring in a general data structure, or analyzing patterns of human behavior to which appropriate program responses must be made in an interactive man-machine system;
2. *Potential for adaptive behavior;* when a significant portion of a program's operation is controlled by a set of rewrite rules, the program's behavior can be changed significantly (for example, in response to training) by automatic program modification of those rules. The stylized form of the rules can be more conducive to automatic modification than more traditional program forms based on procedures, subroutines, and statements governing the flow of program control.

This paper is primarily tutorial in nature. A brief survey of linguistic pattern analysis techniques is followed by an example illustrating the possibilities for adaptive program behavior based on training.

THE LINGUISTIC APPROACH TO PATTERN RECOGNITION

A *language,* such as FORTRAN or English, consists of a set of sentences. (There may be an infinite number of sentences in the set.) Each sentence in this set is constructed according to a collection of *rules,* and each sentence consists of a string of symbols.

For example, consider the sentence

$$S_1 = \text{JOHNbHITbTHEbBALL.}$$

(We denote the symbol ' ', or *blank*, by the symbol 'b').

The following set of rewrite rules might be part of the definition of the language of which S_1 is a member:

<sentence> → <noun phrase> b <verb phrase>.
<noun phrase> → <noun>
<noun phrase> → <article> b <noun>
<verb phrase> → <verb> b <noun phrase>
<noun> → JOHN
<noun> → BALL
<verb> → HIT
<article> → THE

(A character string enclosed in brackets is considered to be a single symbol.) Each rule is of the form

$$c \rightarrow a$$

where *c* is a symbol and *a* is a finite string of symbols.

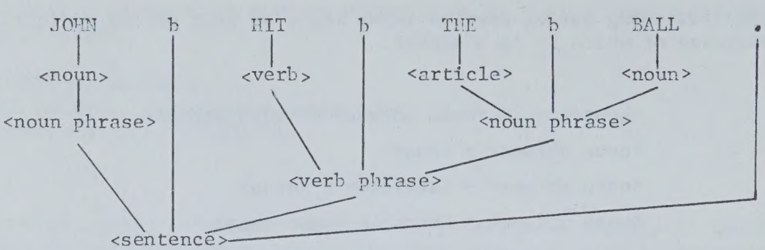
A set of rules defining a language may be used to recognize a sentence in that language. A string of symbols is a sentence only if it is possible to make replacements of rules' right sides occurring within the string by the corresponding left sides so that the resulting string is a single specially designated symbol (in this case, <sentence>).

The following set of replacements show the recognition of "JOHNbHITbTHEbBALL." as a <sentence>. The string of symbols being replaced at each step is underlined.

JOHN b HIT b THE b BALL.
<noun> b HIT b THE b BALL.
<noun phrase> b HIT b THE b BALL.
<noun phrase> b <verb> b THE b BALL.
<noun phrase> b <verb> b <article> b BALL.
<noun phrase>   <verb> b <article> b <noun>.
<noun phrase> b <verb> b <noun phrase>.
<noun phrase> b <verb phrase>.

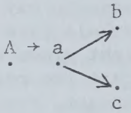
<sentence>

A tree structure shows the replacements clearly:

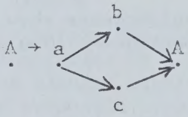


We have demonstrated that a set of rewrite rules can be used to recognize certain patterns (i.e., valid sentences) occurring within a string of symbols. But most patterns of interest do not consist of symbol strings; they consist of data distributed in two or more dimensions, with potentially complex relationships between data elements. One interesting generalization of linguistic concepts, designed to handle such data complexity, uses a labeled, directed graph (or "web") in place of a string of symbols. Sentences in a "web language" are webs, and rewrite rules have webs as their pattern part and replacement part.

For example, consider the following two web rewrite rules:

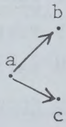


[p1]



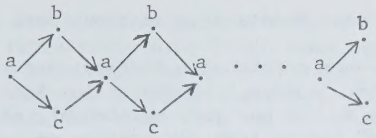
[p2]

Rule p1 intuitively means that an occurrence of a node labeled A within a web may be replaced (using the left side of the rule as the pattern and the right side as the replacement) by the web



However, one must also specify the *embedding* of the right part of rule p1 in place of A within the web; that is, one must specify what to do with edges that end upon A or emanate from A when A is replaced. For the above sample web grammar, the following embedding is assumed: "any edge ending on the A which is being replaced in the host web now ends on the new "a" in the replacement web."

Beginning with a single node labeled "A", rules p1 and p2 generate all webs of the form:



A formal definition of web grammars is contained in Pfaltz and Rosenfeld [5]; that paper also defines "context-sensitive" and "context-free" web grammars and gives examples of each. The theory of web grammars has been extended by Montanari [6] and used in practical recognition systems by Pfaltz [7]. A recent paper by Shaw [8] compares the web grammar notation with other notations for syntactic picture analysis.

Webs are an important generalization of string languages because labeled, directed graphs can be used to represent symbolic objects (labeled nodes or groups of nodes in a graph) and relationships between those objects (labeled arcs between the nodes). The following section contains an example of the use of a labeled, directed graph as a data structure and discusses web rewrite rules as a means of data structure modification and as a basis for adaptive program behavior.

AN EXAMPLE OF PROGRAM ADAPTABILITY

All of the sample grammars illustrated in the previous section were based on the principle that the set of rewrite rules was *unordered*; any rule could be used, if applicable, at any time.

In this section, we consider the use of an *ordered* set of web rewriting rules to govern program behavior. We assume there is a data structure called the *context*, in the form of a labeled, directed graph (i.e., a web) representing the current status of the program. The web rewrite rules are applied according to the following strategy:

> The pattern part of each rule, in order, is tested to see if it matches some subweb of the context. When a match is found, the replacement part of that rule is used to modify the context; then each rule is again tested for applicability to the revised context, in order from the beginning of the set of rules, and the process is repeated until no rule applies to the current context. The system then becomes quiescent until the context is changed by some external agent; at that time the rules are again tested for applicability to the revised context.

(This use of pattern-replacement rules is a restricted form of "programmed grammar" [9].)

Consider the following illustration of the use of an ordered set of web rewriting rules to describe the interpretation of user actions (on a tablet or with a light pen) in an interactive graphical man-machine interface. The user is allowed to create "stations" and directed "flow-lines" between those stations according to the following rules:

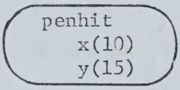
R1: If the user points at a location (x,y) where there is no station, create a station there.

R2: If the user points at a station, mark it "active."

R3: If the user points at a station, and another station is marked "active," create a flow-line from the active station to the one just pointed at; also, remove the "active" marker from the station so marked.

For the purpose of this example, we assume that other parts of the program have mapped an incoming light pen or stylus signal into a component of the context. That component is labeled "penhit;" it has attributes associated with it recording the (x,y) location and links labeled "near" have been created from other components of the context to the penhit if some criterion for "nearness" has been met. (Much of this processing can be accomplished by other web rewrite rules. A fuller explanation of this example and of the adaptive strategies discussed in this paper are contained in a recent research report by Anderson and Sibley [10].)

Figure 1 shows rules R1, R2, and R3 in the form of web rewrite rules. The following notation is used. The symbol


penhit
x(10)
y(15)

stands for a part of the data structure (i.e., it is a subweb of the context web) which represents a logical object -- in this case, "penhit" -- with its

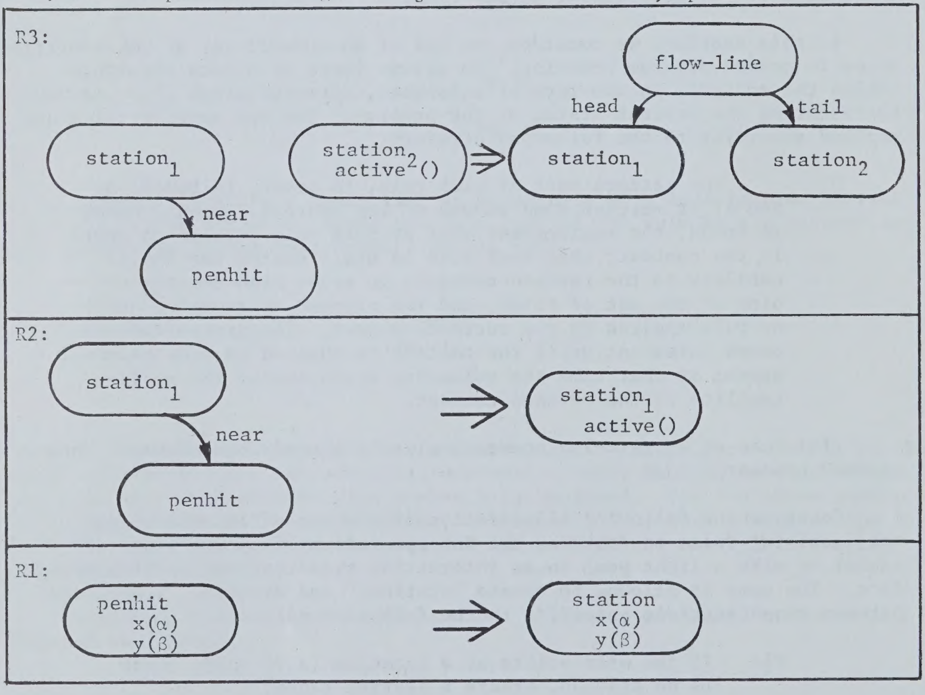

R3:
flow-line
head
tail
station$_1$
station$_2$ active()
station$_1$
station$_2$
near
penhit
R2:
station$_1$
near
penhit
station$_1$ active()
R1:
penhit x(α) y(β)
station x(α) y(β)

Figure 1 -- An Ordered Set of Three Rewrite Rules

associated attributes and their respective values. In the pattern part (left-hand side) and replacement part (right-hand side) of a rewrite rule, the attribute values may be shown as Greek letters; these letters are the names of variables. They are bound to corresponding values in the context during the pattern match, and those same values are used in making the replacement specified by the right-hand side of the rule. Subscripts on the names of logical objects are used to resolve any ambiguity in references to objects appearing on both sides of a rule. A much more rigorous discussion of the operation of this form of web rewrite rule, with extensions to the notation shown here, may be found in [10].

Rules R1, R2, and R3 are shown in Figure 1 in the reverse of the order in which they were originally stated. The ordering in Figure 1 results from the process of "blending" a new rule into an existing set. (Blending is discussed at greater length below.) When R2 is added to the set consisting only of R1, it is placed before R1 by the algorithm for blending; similarly, R3 is placed before R2 and R1. Intuitively, this is because the pattern part of R2 has more stringent requirements than the pattern of R1, and R3 is more stringent than R2.

The rewrite rules in Figure 1 allow the interpretation of user actions so that a network of stations may be constructed whose topology is recorded in the data structure. (We assume each logical object, such as "station" and "flow-line," also has an ouput representation on a display device so that the network may be visualized as it is being constructed.)

But the three rules in Figure 1 do not handle all cases reasonably. If a user points twice in succession to an existing station, that station will be given two "active()" attributes by the action of rule R2. Assume that program behavior is discovered during operation of the program, and a different interpretation is to be given to those actions: the first penhit to an existing station makes it "active," but if that same station is hit again, the "active" attribute is to be removed. Rule R4 in Figure 2 embodies this logic.

To modify the existing program behavior, Rule R4 is blended into the existing set of rules. The algorithm for blending a new rule that corrects faulty program behavior into an existing set of rules is the following:

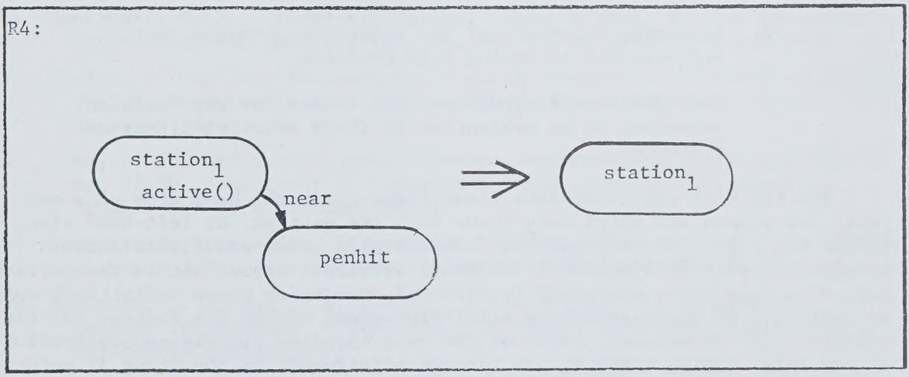


Figure 2 -- A Fourth Rewrite Rule to be Blended into the Existing Set

1. Locate the error-causing rule; i.e., the rewrite rule responsible for the unacceptable decision. (In the example, rule R2 is the "error-causing" rule; it adds "active()" attributes to a station that is hit twice in succession.)

2. Search the rules above the error-causing rule for a rule that can be modified to incorporate the new rule. To be eligible for such modification a rule must have the same right-hand side as the new rule (within a mapping of variable names). (In the above example, rule R3 is the only rule above R2; it does not meet this criterion.)

3. Search the rules below the error-causing rule for a rule that can be modified to incorporate the new rule. If one is found -- call it Ri -- and if the error-causing rule and the rules between it and Ri can be modified to pass the error-causing situation, then perform all of these modifications. (In the example, R1 cannot be so modified.) The exact nature of the modifications requires more explanation than can be given here; see [10] for a complete description.

4. If none of the above criteria are met, place the new rule immediately above the error-causing rule. (In the example, R4 is therefore placed between R3 and R2.)

The process of blending a new rule into an existing set, outlined above, is based on a discussion by Waterman [11] of adaptivity in rule-directed programs. (Also, see Evans for a discussion of adaptivity in grammar-controlled programs [2,12].) As Waterman points out, a new rule modifying program behavior can be constructed from three items of information provided by the user in a training situation:

1. *Acceptability information;* an acceptable program response for the situation.

2. *Relevancy information;* the situation elements that are relevant to making this decision.

3. *Justification information;* the reason for the decision, expressed as an evaluation of these relevant situation elements.

The first of the above data items forms the right-hand side of a new rule; the second and third data items form the pattern, or left-hand side, of the new rule. In [10] and [11] considerable power and flexibility is attained in rule definitions by allowing attribute values within the pattern part of a rule to be expressed in terms of *categories* whose definitions must be satisfied by the appropriate attribute values within the context for the pattern match to succeed. Much of the rule blending process and definition of new rules during training can involve adjustments to the scope of category definitions.

CONCLUSION

The pattern-replacement rules used in traditional character-string-oriented grammars can be generalized to allow labeled, directed graphs as their pattern- and replacement-part. Ordered sets of such rules can control the manipulation of a data structure; these sets of rules can be modified in a training situation to allow program behavior to adapt to the training information. The resulting program flexibility is both broadly applicable and significant in its extent.

ACKNOWLEDGMENT

Much of the material in this paper is derived from a research project conducted jointly by the author and William L. Sibley of The Rand Corporation, Santa Monica, California. The author wants to acknowledge the substantial contributions made by Mr. Sibley both to that project and to the research report co-authored by him on which much of this paper is based.

REFERENCES

1. Miller, W. F., and A. C. Shaw, "Linguistic Methods in Picture Processing -- A Survey," Proceedings of the Fall Joint Computer Conference (1968), pp. 279-290.
2. Evans, Thomas G., "Grammatical Inference Techniques in Pattern Analysis," *Software Engineering*, Vol. 2 (Tou, Julius T., ed.), Academic Press, New York (1971), pp. 183-202.
3. Fu, K. S., and P. H. Swain, "On Syntactic Pattern Recognition," *Software Engineering*, Vol. 2, (Tou, Julius T., ed.), Academic Press, New York (1971), pp. 155-182.
4. Uhr, L., "Flexible Linguistic Pattern Recognition," *Pattern Recognition*, Vol. 3, No. 4 (November 1971), pp. 363-383.
5. Pfaltz, J. L. and A. Rosenfeld, "Web Grammars," Proceedings of the International Conference on Artificial Intelligence, Washington, D.C. (May 1969), pp. 609-619.
6. Montanari, U. T., "Separable Graphs, Planar Graphs, and Web Grammars," *Information and Control*, Vol. 16, No. 3 (May 1970), pp. 243-267.
7. Pfaltz, J. L., "Web Grammars and Picture Description," Technical Report 70-138, Computer Science Center, University of Maryland, College Park, Maryland (September 1970).
8. Shaw, A. C., "Picture Graphs, Grammars, and Parsing," Technical Report No. 71-89, Department of Computer Science, Cornell University, Ithaca, New York (January 1971).
9. Rosenkrantz, Daniel J., "Programmed Grammars and Classes of Formal Languages," *JACM*, Vol. 16, No. 1 (January 1969), pp. 107-131.
10. Anderson, R. H. and W. L. Sibley, "A New Approach to Programming Man-Machine Interfaces," R-876-ARPA, The Rand Corporation, Santa Monica, California (March 1972).

11. Waterman, D. A., "Generalization Learning Techniques for Automating the Learning of Heuristics," *Artificial Intelligence,* Vol. 1 (1970), pp. 121-170.

12. Evans, Thomas G., "A Grammar-Controlled Pattern Analyzer," *Proc. IFIP Congress 68,* North-Holland Publishing Company, Amsterdam (1969), pp. 1592-1598.

ON THE AUTOMATIC GENERATION OF HYPOTHESIS

R. C. T. Lee and C. L. Chang
Division of Computer Research and Technology, National Institutes of Health, Bethesda, Maryland 20014

ABSTRACT

In this paper, we discuss the following problem: Given a set of observed facts, how can we find a set of hypothesis to explain these facts? If we assume that the observed facts can be described by a logical formula, then the above problem becomes the consequent finding problem. It is shown that we can define a language, called the E-standard form, and an inference rule, called the I-resolution principle, for the purposes of generating logical consequents. It is proved that the E-standard form of a formula preserves the validity of that formula and the I-resolution principle is complete in generating prime logical consequents.

1. INTRODUCTION

In this paper, we are interested in the following problem: Given a set of observed facts, how can we find a set of hypothesis to explain these facts?

If we assume that the set of facts can be described by a logical formula B, then the above problem becomes the following problem: Given a logical formula B, how can we find a logical formula A such that A implies B? Those who are familiar with symbolic logic will note that, in this case, formula A is sometimes called an axiom and formula B is called the logical consequence of A.

Thus the hypothesis finding problem is similar to the consequence finding problem. In the consequence finding problem, we are given a formula A and we are asked to find a formula B such that A implies B.

The hypothesis finding (or axiom finding) problem has been considered by Meltzer [1970] and Morgan [1971]. The consequence finding problem has been investigated by Lee [1967], Slagle, Chang and Lee [1969] and Slagle [1970]. Since these two problems are highly related, we shall first briefly introduce some progress made in the consequence finding problem.

It is assumed that the reader is reasonably familiar with symbolic logic and mechanical theorem proving. For a comprehensive study of this problem, see [Slagle 1971], [Nilsson 1971] and [Chang and Lee 1973].

2. THE CONSEQUENCE FINDING PROBLEM

Let us first define logical consequence.

Definition. A formula B is a logical consequence of a formula A if and only if for every interpretation I in which A is true, B is also true.

There are two basic elements in designing an inference system: the language and the inference rule. We shall discuss these two items now.

(a) The language: In the case of deducing logical consequces, we shall use the standard form introduced in [Chang and Lee 1973].

Roughly speaking, the standard form of a formula A is obtained according to the following procedure:

Procedure to obtain a standard form of a formula A:

(1) Transform A into prenex normal form $(Q_1x_1)\ldots(Q_nx_n)M$ where every Q_i is a quantifier, universal or existential, and M is a quantifier-free formula in conjunctive normal form.

(2) Suppose Q_r is an existential quantifier in the prefix $(Q_1x_1)\ldots(Q_nx_n)$, $1\leq r\leq n$. If no universal quantifier appears before Q_r, we choose a new constant c different from other constants occurring in M, replace all x_r's appearing in M by c and delete (Q_rx_r) from the prefix. If $Q_{s_1}, \ldots, Q_{s_r}$ are all the universal quantifiers before Q_r, $1\leq s_1\leq s_2 \ldots \leq s_m < r$, we choose a new m-place function symbol f different from other function symbols, replace all x_r's in M by $f(x_{s_1},\ldots,x_{s_m})$ ($f(x_{s_1},\ldots,x_{s_m})$ is called a Skolem function.) and delete (Q_rx_r) from the prefix.

(3) After the above process is applied to all the existential quantifiers in the prefix, the last formula we obtain is called the standard form of the formula A.

For some reason that will become clear later, we shall call the above standard form the U-standard form because every variable that it contains is interpreted as universally quantified.

Example 1.

Consider the following formula A:
$(x)(y)(P(x,y)\longrightarrow P(y,x))\&P(a,b)$.
A U-standard form of A is
$(x)(y)((\sim P(x,y)vP(y,x))\&P(a,b)$.

(b) The inference rule:

Once we obtain the U-standard form of a formula, we have essentially obtained a set of clauses and we can now use a famous mechanical inference rule, called the resolution principle [Robinson 1965, Slagle 1971, Nilsson 1971, Chang and Lee 1973] to deduce logical consequences.

Example 2.

In Example 1, we have two clauses:

C_1 : $\sim P(x,y)vP(y,x)$

C_2 : $P(a,b)$

From C_1 and C_2, by using the resolution principle, we can now obtain

C_3 : $P(b,a)$.

But, obviously, there are an infinite number of clauses that are logical consequences of a set of clauses. For instance, consider the clause

C : R

We can see that all of the following clauses are logical consequences of C:

C_1 : RvQ_1

C_2 : RvQ_2

C_3 : RvQ_3

.

.

.

We thus face the following problem: Given a set S of clauses, can we use the resolution principle to generate all of the logical consequences of S? The answer to the above question is "No", because it is evident that none of the clauses mentioned above can be deduced by using the resolution principle. But, it was proved that resolution principle is complete in deducing "prime logical consequences" [Lee 1967, Slagle, Chang and Lee 1969].

Definition. A clause C is a prime logical consequence of a set S of clauses if and only if C is a logical consequence of S and there exists no other clause C' such that C' is a logical consequence of S and C is a logical consequence of C'.

Example 3.

Consider

$$S : \begin{cases} \sim L_1 v L_2 \\ L_1 v L_3 \end{cases}$$

Clause

C_1 : $L_2 v L_3$

is a logical consequence of S and it can be proved that C_1 is also a prime logical consequence of S. The reader can easily see that the clause

C_2 : $L_2 v L_3 v L_4$

is a logical consequence of S, but not a prime logical consequence of S because C_2 is also a logical consequence of C_1.

The main result of [Lee 1967] shows that, given a set S of clauses, such clause as C_2 may not be deduced by resolution principle, but clause C_1, which is a prime logical consequence, will definitely be deduced by repeatedly applying the resolution principle. We restate the main theorem of [Lee 1967] as follows.

Theorem 1. Given a set S of clauses, every prime logical consequence (in the form of clauses) of S will be deduced by using the resolution principle.

Theorem 1 shows the power of resolution principle to deduce logical consequences. Although it may fail to deduce some of the logical consequences, these logical consequences must be implied by some other logical consequences that will be generated by using the resolution principle.

We have completed a brief introduction of the consequence finding problem. In the following sections, we shall talk about how to find hypothesis. In this case, instead of deducing logical consequences, we induce logical consequents. Instead of using the U-standard form of a formula, we use the E-standard form. Instead of using the resolution principle, we shall use the I-resolution principle.

3. The E-standard Form, I-resolution Principle and Inducing Logical Consequents

Let us first define logical consequent.

Definition. A formula A is a logical consequent of a formula B if and only if B is a logical consequence of A.

To deduce logical consequences, we transform a formula into a conjunction of clauses where each clause is a disjunction of literals. To induce logical consequents, we can transform a formula into a disjunction of conjunction of literals.

Definition. A phrase is a conjunction of literals.

Based on two phrases, we can now define an I-resolvent.

Definition. Given two phrases P_1 and P_2, suppose that P_1 and P_2 are of the form of $L_1\&L_2\ldots\&L_m$ and $\sim L_1\&L_2'\ldots\&L_n'$ respectively, then $L_2\&\ldots\&L_m\&L_2'\ldots\&L_n'$ is an I-resolvent of P_1 and P_2.

Example 4.

Consider the following two phrases:

P_1 : $\sim L_1\&L_2$

P_2 : $L_1\&L_3$.

We can induce an I-resolvent

P : $L_2\&L_3$

out of P_1 and P_2.

Lemma 1. The I-resolvent P of two phrases P_1 and P_2 is a logical consequent of $P_1 \vee P_2$.

The proof of this lemma is omitted.

In the first order predicate logic, there are two kinds of variables : universal and existential. In order to apply the I-resolution principle, we have to eliminate one of them.

Let us consider the following two phrases:

$\sim L_1(x)\&L_2(x)$ P_1

$L_1(a)$ P_2

From the above two phrases, we can induce

$L_2(a)$ P

as an I-resolvent of P_1 and P_2.

Case 1. Assume variable x is interpreted as universally quantified.

We can now prove that P is not a logical consequent of $P_1 v P_2$. This can be done by considering the following interpretation:

$D = \{a, b\}$.

$L_1(a) \longleftarrow F$, $L_1(b) \longleftarrow T$, $L_2(a) \longleftarrow T$, $L_2(b) \longleftarrow F$.

Obviously, P is true in I. But

$(\sim L_1(b) \& L_2(b)) \& L_1(a)$ is false in I. Therefore

$(x)(\sim L_1(x) \& L_2(x)) v L_1(a)$ is false in I and P is not a logical consequent of $P_1 v P_2$.

Case 2. Suppose that variable x is interpretated existentially quantified.

Note that in every interpretation I in which P is true, $L_2(a)$ is true. It can be easily shown that no matter how we assign the truth-value of $L_1(a)$, $P_1 v P_2$ will be true in I. Therefore P is a logical consequent of $P_1 v P_2$.

To eliminate the universal quantifiers, we may use the Skolem function introduced in Section 2. In fact, if we replace "universal", "existential", "conjunctive" and "disjunctive" by "existential", "universal", "disjunctive" and "conjunctive" respectively in the procedure to obtain the U-standard form of a formula, then the formula we obtain will be called the E-standard form of a formula (meaning that every variable in the formula is an existential quantifier).

Example 5.

Consider

$(x)(\exists y)(\exists z)(u)(\exists v)(w)P(x,y,z,u,v,w)$.

Since (x) is not preceded by any existential quantifier, it is replaced by a constant, say "a". (u) is preceded by $(\exists y)$ and $(\exists z)$. It is replaced by $f_1(y,z)$. Similarly, (w) will be replaced by $f_2(y,z,v)$. The original formula is now

$P(a,y,z,f_1(y,z),v,f_2(y,z,v))$.

We have defined the language and the inference rule for inducing logical consequents. We now ask the following questions:

(a) Let A' denote the E-standard form of a formula A. Is A' equivalent to A? If not, what kind of property of A does A' still preserve?

(b) Is the I-resolution principle complete in inducing logical consequents?

In the following sections, we shall answer the above two questions.

4. SOME PROPERTIES OF THE E-STANDARD FORMS

Let us first answer the following question:

Is the E-standard form of a formula A equivalent to A?

The answer is "No". To see this, consider

A: $(x)P(x)$.

The E-standard form of A is

A': $P(a)$.

Evidently, A implies A', but A' does not imply A. Therefore A and A' are not equivalent.

But it is interesting to note that although the E-standard form of a formula A is not equivalent to A, it will still preserve the validity property of A. This can be seen from the following theroem.

Theorem 2. Let F be a formula in the first order logic. Let F' be an E-standard form of F. Then F is valid if and only if F' is valid.

Proof:

Without loss of generality, we may assume that F is in a prenex normal form, i.e.,

$$F=(Q_1x_1)\ldots(Q_nx_n)M[x_1,x_2,\ldots,x_n].$$

(We use $M[x_1,x_2,\ldots,x_n]$ to denote that the matrix M contains variables $x_1, x_2, \ldots, x_n$.) Let Q_r be the first universal quantifier. Let $F_1=$

$$(\exists x_1)(\ldots(\exists x_{r-1})(Q_{r+1}x_{r+1})\ldots(Q_nx_n)M[x_1,\ldots x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$$

where f is a Skolem function corresponding to x_r, $1\leq r\leq n$. We now show that F is valid if and only if F_1 is valid.

(a)Suppose F is valid and F_1 is not valid. Then there exists an interpretation I such that F_1 is false in I. That is, there exist $x_1,x_2,\ldots,x_{r-1}$ and an element $f(x_1,\ldots,x_{r-1})$ such that $(Q_{r+1}x_{r+1})\ldots(Q_nx_n)M[x_1,\ldots,x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$ is false in I. This means that

$$(\exists x_1)\ldots(\exists x_{r-1})(x_r)(Q_{r+1}x_{r+1})\ldots(Q_nx_n)M[x_1,\ldots,x_r,\ldots,x_n]$$ is false in I. Or, equivalently, F is false in I which contradicts the assumption that F is valid. Therefore, F_1 must be valid.

(b) Suppose F_1 is valid. In this case, F_1 is true in every interpretation over any domain D. That is, no matter how the function f maps $(x_1,\ldots,x_{r-1})$ onto some element in D, $(\exists x_1)(\ldots(\exists x_{r-1})(Q_{r+1}x_{r+1})\ldots(Q_nx_n)$ $M[x_1,\ldots,x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$ is true in D. Therefore, there exist $x_1, \ldots, x_{r-1}$ such that for all x_r, $(Q_{r+1}x_{r+1})\ldots(Q_nx_n)M[x_1,\ldots,$

$x_{r-1}, x_r, x_{r+1}, \ldots, x_n]$ is true in every I. This means that F is valid.

Assume that there are m universal quantifiers in F. Let $F_0=F$. Let F_k be obtained from F_{k-1} by replacing the first existential quantifier in F_{k-1}^k by a Skolem function, k=1,...,m. Clearly, $F'=F_m$. Using the same arguments given above, we can show that F_{k-1} is valid if and only if F_k is valid for k=1,...,m. Therefore, we conclude that F is valid if and only if F' is valid. Q.E.D.

5. THE COMPLETENESS OF THE I-RESOLUTION PRINCIPLE IN INDUCING LOGICAL CONSEQUENTS

In this section, we shall answer the second question: Is the I-resolution principle complete in inducing logical consequents?

Before answering this question, let us note that many important concepts in theorem proving are still useful here. First of all, the reader can easily prove that in order to show the validity of a set of phrases, it is only necessary to consider interpretations defined over Herbrand universes. (Consult Chapter 4 of [Chang and Lee 1973] for this problem.) Secondly, the lifting lemma of the resolution principle is also true for the I-resolution principle. That is, if P_1' and P_2' are instances of phrases P_1 and P_2 respectively, then there is an I-resolvent P of P_1 and P_2 such that P' is an instance of P. For a proof of this lemma, consult Chapter 5 of [Chang and Lee 1973].

We now prove the completeness of the I-resolution principle for inducing logical consequents.

Theorem 3. Given a set S of phrases, if a phrase P is a logical consequent of S, then there is an I-resolution induction of either P or a phrase P' implied by P.

Proof:

Let $P \triangleq L_1 \& L_2 \ldots \& L_n$. Let A denote the set of all the Herbrand instances over the set of all the distinct atomic formulas occurring in S and P, or whose complements occur in S and P. Let G be the set of all the distinct Herbrand instances of $L_1, \ldots, L_n$. G evidently satisfies P. Besides, every interpretation satisfying P must contain G.

Let $B=\{B_1, B_2, \ldots, B_j, \ldots\}=A-G$. We now construct a semantic tree T of S as shown in Fig. 1. For every node N of the tree, let I(N) denote the set of literals attached to the path from the root of the tree to node

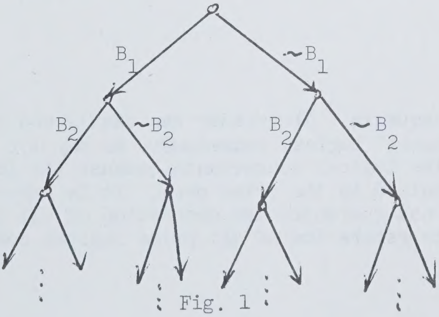


Fig. 1

N. Whenever for some N, $I(N) \cup G$ satisfies a phrase in S, we terminate the node N and denote this node a failure node. Since P implies S, every interpretation that satisfies P satisfies at least one phrase in S. Therefore, every path of the tree will be finally terminated by a failure node.

Let N_{12} be a node of the semantic tree T which is not a failure node, but whose immediate successors are failure nodes. Let N_1 and N_2 be these nodes. There must be two ground instances P_1' and P_2' of phrases P_1 and P_2 such that P_1' and P_2' are true in $I(N_1)$ and $I(N_2)$ respectively, but both P_1' and P_2' are not satisfied by $I(N_{12})$. Therefore we can obtain an I-resolvent P_{12}' of P_1' and P_2'. P_{12}' must be true in $I(N_{12})$. By the lifting lemma, there must be an I-resolvent P_{12} of P_1 and P_2 such that P_{12}' is a ground instance of P_{12}. Let T' be the semantic tree for $(S \cup \{P_{12}\})$ obtained from T by deleting any node and link which is below the first node where the I-resolvent P_{12} is satisfied. The number of nodes in T' is less than that in T. Applying the process on T' again, we can obtain another I-resolvent of phrases in $(S \cup \{P_{12}\})$. Putting this I-resolvent into $(S \cup \{P_{12}\})$, we get another closed semantic tree. We repeat this process until a semantic tree consisting of only the root node is generated. Let P' denote the final I-resolvent generated. P' must be satisfied by G. Therefore, either P' is P itself or P' is implied by P. Q.E.D.

We may define a prime logical consequent as follows:

Definition. A phrase P is a prime logical consequent of a set S of phrases if and only if P is a logical consequent of S and if there exist no other phrases P' such that P' is a logical consequent of S and P is a logical consequent of P'.

Using the concept of prime logical consequents, we can restate Theorem 3 in the following form.

Theorem 4. Let S be a set of phrases. Every prime logical consequent of S can be induced by some I-resolution induction.

Let us examine deeper into the meaning of prime logical consequents. Suppose

$$P\&Q$$

is a logical consequent of a set S of phrases. Then, certainly

$$P\&Q\&R_1$$

$$P\&Q\&R_1\&R_2$$

.
.
.

are all logical consequents. The reader can easily see that, in some sense, P&Q is the most "compact" logical consequent. We may not care whether we can generate the non-prime logical consequents because the information that they contain is also contained in the prime ones. It is marvelous that although the generation does not guarantee the generation of all logical consequents, it does guarantee the generation of all prime logical consequents.

References

1. Chang, C. L. and Lee, R. C. T. (1973) : *Symbolic Logic and Mechanical Theorem Proving*, to be published by Academic Press, N.Y., 1973.

2. Lee, R. C. T. (1967) : *A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms*, Ph.D. Thesis, University of California, Berkeley, Calif. 1967.

3. Meltzer, B. (1970) : The semantics of induction and the possibility of complete systems of inductive inference, *Artificial Intelligence*, Vol. 1, No. 3, 1970, pp.189-192.

4. Morgan, C. G. (1971) : Hypothesis generation by machine, *Artificial Intelligence*, Vol. 2, 1971, pp.179-187.

5. Nilsson, N. J. (1971) : *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill, N.Y., 1971.

6. Robinson, J. R. (1965) : A machine-oriented logic based on the resolution principle, *Journal of the ACM*, Vol. 12, No. 1, Jan. 1965, pp.23-41.

7. Slagle, J. R. (1970) : Interpolation theorems for resolution in lower predicate calculus, *Journal of the ACM*, Vol. 17, No. 3, July 1970, pp.535-542.

8. Slagle, J. R. (1971) : *Artificial Intelligence, the Heuristic Programming Approach*, McGraw-Hill, N.Y., 1971.

9. Slagle, J. R., Chang, C. L. and Lee, R. C. T. (1969) : Completeness theorem for semantic resolution in consequence finding, *Proc. of the First International Joint Conference on Artificial Intelligence*, 1969, pp.281-285.

THE DECOMPOSITION PRINCIPLE FOR THEOREM PROVING SYSTEMS

CHIN-LIANG CHANG
Division of Computer Research and Technology, National Institutes of Health, Bethesda, Maryland 20014

ABSTRACT

Given a theorem, represented by a set of clauses, to be proved, the idea of the decomposition principle is to first break the theorem into small and simple cases and then prove the theorem by considering each case separtely. This paper describes a method to break up the theorem. Proving a theorem by using this method may yield a great savings.

1. Introduction

Before I go into the main topic of this talk, let me first briefly describe what a theorem-proving system is. Roghly speaking, a theorem-proving system consists of a (formal) language and a set of "inference" rules for the language. In this paper, the language I am considering is the language of the first order logic. The reason for using this language is that deduction can be performed in this language. In this language, we have the following types of symbols:

(1) constant symbols,
(2) variable symbols,
(3) function symbols,
(4) predicate symbols,
(5) quantifier symbols: $\forall$ and $\exists$,
(6) symbols of logical connectives: $\sim$, &, v, $\rightarrow$ and $\leftrightarrow$.

Using these symbols, we can construct sentences or forumlas in the language [4]. Suppose $A_1,\ldots,A_n$ and B are formulas. The essential job of a theorem-proving system is to prove whether or not B can be deduced from $A_1,\ldots,A_n$. Let us consider a theorem-proving example.

Example 1. The premise is that everyone who saves money earns interest. The conclusion is that if there is no interest, then noboby saves money. Let S(x,y), M(x), I(x) and E(x,y) denote "x saves y", "x is money", "x is interest" and "x earns y", respectively. Then, the premise can be represented by the formula A given as

$$A \triangleq (\forall x)((\exists y)(S(x,y)\&M(y)) \rightarrow (\exists y)(I(y)\&E(x,y))).$$

The conclusion can be represented by the formula B given as

$$B \triangleq \sim(\exists x)I(x) \longrightarrow (\forall x)(\forall y)(S(x,y) \longrightarrow \sim M(y)).$$

The problem now is to prove that B can be deduced from A.

A theorem-proving system can be used in many other systems. For example, consider an English question-answering system as shown in Fig. 1.

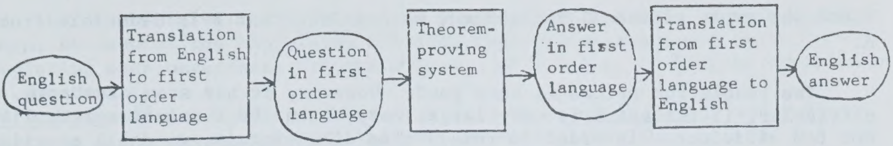

English question
Translation from English to first order language
Question in first order language
Theorem-proving system
Answer in first order language
Translation from first order language to English
English answer

Figure 1. An English question-answering system

In this system, the theorem-proving system can provide deductive power for the question-answering system. There are many other systems, e.g., problem-solving systems, chemical synthesis systems, program-writing systems, etc., in which a theorem-proving system can be used. If you have a system to build, first try to see whether or not your system can be described (represented) by formulas in the first order language. If yes, then a theorem-proving system may be used as a package in your system.

Because of the role which a theorem-proving system could play in many other systems, developing an efficient theorem-proving system is fundamentally important. That is, given formulas $A_1,\ldots,A_n$ and B, we shall now consider good ways to prove whether or not B can be deduced from $A_1,\ldots,A_n$ by using inference rules provided for the language. There are many proof techniques proposed to do this job. The most powerful computer-oriented technique so far is the resolution method introduced by J.A. Robinson [8]. Since it can be shown that B is deducible from $A_1,\ldots,A_n$ if an only if $A_1\&\ldots\&A_n\&\sim B$ is unsatisfiable, Robinson's method is to verify the latter. Let a clause be disjunction of literals. His method is to first transform $A_1\&\ldots\&A_n\&\sim B$ into a set S of clauses, and then use an inference rule called "the resolution principle" to generate new clauses from clauses in S. It is shown that $A_1\&\ldots\&A_n\&\sim B$ is unsatisfiable, or B is deducible from $A_1,\ldots,A_n$, if and only if the empty clause $\square$ can be deduced from S.

Example 2. Consider the formulas A and B in Example 1. $A\&\sim B$ can be transformed into the following clauses:

(1) $\sim S(x,y) \vee \sim M(y) \vee I(f(x))$
(2) $\sim S(x,y) \vee \sim M(y) \vee E(x,f(x))$
(3) $\sim I(z)$
(4) $S(a,b)$
(5) $M(b)$

Using resolution, we can obtain the following deduction:

(6)	$\sim S(x,y) \vee \sim M(y)$	a resolvent of clause (3) and clause (1)
(7)	$\sim M(b)$	a resolvent of clause (6) and clause (4)
(8)	$\square$	a resolvent of clause (7) and clause (5)

Since the empty clause $\square$ is deduced, we conclude that B is deducible from A.

The resolution method is very good. However, it has soem drawbacks. In particular, if the set S is very large, very often the resolution method is not too efficient. In order to remedy this disadvantage, we shall consider a decomposition method. The idea is to first break S into small and simple cases and then prove that S is unsatisfiable by considering each case separately. Several decomposition methods have been proposed by Prawitz [6,7], Bledsoe [1], Slagle and Koniver [9], Ernst [5], and Wos, Robinson, Carson, and Shalla [10]. In this paper, we shall consider a decomposition method based on V-resolution [3].

2. V-Resolution

Let S be a set of caluses to be proved unsatisfiable. Suppose $C=C_1 \vee C_2$ is a clause in S. Let

$S_1 = \{ C_1 \} \cup (S - \{ C \})$ and

$S_2 = \{ C_2 \} \cup (S - \{ C \})$.

Now, can we claim that S is unsatisfiable if both S_1 and S_2 are unsatisfiable? The answer can be put in this way: Yes, if C_1 and C_2 have no variables in common; no, if C_1 and C_2 have variables in common. It is easy to see that both S_1 and S_2 are simpler than S and thus easier to prove. Therefore, it would be advantageous to break S into S_1 and S_2. However, if C_1 and C_2 have variables in common, C can not be split. To overcome this difficulty, we change those variables which are common in C_1 and C_2 into parameters. Parameters will be considered to be unknown terms which are to be found. After those variables which occurs in both C_1 and C_2 are parameterized, S is decomposed into S_1 and S_2. Now, to prove that S is unsatisfiable, we can prove both S_1 and S_2 are unsatisfiable. Since S_1 and S_2 may contain variables and parameters, we shall use V-resolution to prove them. A detailed description of our decomposition method will be given in the next section. In the remainder of this section, we first consider the following defintions. Most of these definitions can be found in [3,4].

Definition. A substitution is a finite set of the form $\{ t_1/v_1, \ldots, t_n/v_n \}$ which satisfies the following conditions:

(1) for $i=1,\ldots,n$, v_i is either a parameter or a variable, and t_i is a term different from v_i;

(2) for $i=1,\ldots,n$, if v_i is a parameter, then t_i must be a term containing no variables;

(3) No two elements in the set have the same variable after the stroke symbol.

The above definition differs from the definition given in [8] by having the additional condition (2). That is, a parameter can only be replaced by a constant, another parameter, or a term containing no variable. The empty substitution is always denoted by ε.

Let $\Theta = \{ t_1/v_1,\ldots,t_n/v_n \}$ be a substitution and E be an expression. Then, $E\Theta$ denotes the expression obtained from the expression E by replacing each occurrence the variable v_i ,$1 \leqslant i \leqslant n$, in E by an occurrence of the term t_i.

Definition. A set W of expressions is said to be unifiable if and only if there is a substitution Θ such that $W\Theta$ is a singleton. Such a Θ is called a unifier for W.

Given a set W of expressions, the folloiwng algorithm can find a most general unifier for W if W is unifiable. If W is not unifiable, the algorithm can also detect this fact.

Unification Algorithm:

Step 1. Set k=0, $W_k=W$ and $\sigma_k = \varepsilon$.

Step 2. If W_k is a singleton, stop; σ_k is a most general unifier for W. Otherwise, find the disagreement set D_k of W_k.

Step 3. If there are elements v_k and t_k in D_k such that v_k is a variable which does not occur in t_k, go to Step 6. Otherwise, continue.

Step 4. If there are elements v_k and t_k in D_k such that v_k is a parameter which does not occur in t_k and t_k contains no variable, go to Step 6.
Otherwise, continue.

Step 5. If there are elements v_k and t_k in D_k such that v_k is a parameter which does not occur in t_k and t_k contains variables, say $x_1,\ldots,x_m$, to to Step 7. Otherwise,stop; W is not unifiable.

Step 6. Let $\sigma_{k+1} = \sigma_k\{t_k/v_k\}$,$W_{k+1}=W_k\{t_k/v_k\}$ and go to Step 8.

Step 7. Choose new parameters, say $p_1,\ldots,p_m$, which have never been used
Let $\sigma_{k+1} = \phi_k \{p_1/x_1,\ldots,p_m/x_m\}$ and $W_{k+1}=W_k\{p_1/x_1,\ldots,p_m/x_m\}$.

Step 8. Set k=k+1 and go to Step 2.

We can now consider an inference rule called V-resolution. V-resolution is an abbreviation for variable-constrained resolution where parameters are considered as constrained variables. Since substitutions for parameters must be consistent, we first give the following definition.

Definition. Let $\Theta_1 = \{ s_1/u_1,\ldots,s_m/u_m \}$ and $\Theta_2 = \{t_1/v_1,\ldots,t_n/v_n\}$ be two substitutions for parameters, i.e., $u_1,\ldots,u_m,v_1,\ldots,v_n$ are all parameters. From Θ_1 and Θ_2, we define two expressions by $E_1=(u_1,\ldots,u_m,v_1,\ldots,v_n)$ and $E_2=(s_1,\ldots,s_m,t_1,\ldots,t_n)$. The Θ_1 and Θ_2 are said to be consistent if and only if E_1 and E_2 are unifiable. A most general unifier for $\{ E_1,E_2 \}$ is called a combination of Θ_1 and Θ_2. The substitutions Θ_1 and Θ_2 are said to be inconsistent if and only if they are not consistnet.

In order to keep track of substitutions for parameters, a clause will alsways be represented by $C|\Theta$, where C is a disjunction of literals and Θ is a substitution for parameters. Each original clause will be represented by a disjunction of literals followed by the empty substitution ε. We now define V-resolution which is also given in [3,4].

Definition. If two or more literals (with the same sign) of a clause $C|\Theta$ have a most general unifier $\sigma = \alpha \cup \beta$, where α and β are substitutions for variables and parameters respectively, and if Θ and β are consistent, with $\emptyset$ being a combination of Θ and β, then $C\sigma|\emptyset$ is called a V-factor of $C|\Theta$.

Definition. Let $C_1|\Theta_1$ and $C_2|\Theta_2$ be two clauses with no variables in common. If Θ_1 and Θ_2 are consistent, with $\emptyset$ being a combination of Θ_1 and Θ_2, if there are literals L_1 and L_2 in C_1 and C_2 respectively such that L_1 and $\sim L_2$ have a most general unifier $\sigma = \alpha \cup \beta$, where α and β are substitutions for variables and parameters respectively, and if $\emptyset$ and β are consistent, with Θ being a combination $\emptyset$ and β, then we infer the clause $(C_1' \cup C_2')|\Theta$, where C_i' is a disjunction of literals obtained from $C_i\sigma$ by deleting $L_i\sigma$, $i=1,2$. The above clause is called a binary V-resolvent of $C_1|\Theta_1$ and $C_2|\Theta_2$. L_1 and L_2 are called the literals resolved upon.

Definition. A V-resolvent of clauses $C_1|\Theta_1$ and $C_2|\Theta_2$ is either of the following binary V-resolvents:

(1) a binary V-resolvent of $C_1|\Theta_1$ and $C_2|\Theta_2$;

(2) a binary V-resolvent of $C_1|\Theta_1$ and a V-factor of $C_2|\Theta_2$;

(3) a binary V-resolvent of a V-factor of $C_1|\Theta_1$ and $C_2|\Theta_2$;

(4) a binary V-resolvent of a V-factor of $C_1|\Theta_1$ and a V-factor of $C_2|\Theta_2$.

We note that V-resolution is the same as resolution except that we have to check the consistency of substitutions for parameters. When no parameters are in $C_1|\Theta_1$ and $C_2|\Theta_2$, a V-resolvent of $C_1|\Theta_1$ and $C_2|\Theta_2$ is just a resolvent.

3. A Decomposition Method

Using V-resolution, we can now describe a procedure for decomposing a set S of clauses as follows.

A Decomposition Procedure:

Step 1. Choose a nonunit clause $C|\varepsilon$ in S. (Note that since all clauses in S are original clauses, each of them is followed by ε.)

Step 2. Let $C=C_1 \vee C_2$.

Step 3. Find variables, say $x_1,\ldots,x_m$, which appear in both C_1 and C_2.

Step 4. Select new parameters, say $p_1,\ldots,p_m$, which have never been used before.

Step 5. Let C_1^* and C_2^* be obtained from C_1 and C_2 respectively by changing x_i to p_i, $i=1,\ldots,m$.

Step 6. Define

$$S_1 = \{ C_1^* | \varepsilon \} \cup (S - \{ C | \varepsilon \})$$

$$S_2 = \{ C_2^* | \varepsilon \} \cup (S - \{ C | \varepsilon \})$$

Step 7. Use V-resolution to attempt to derive empty clauses $\square | \theta_1$ and $\square | \theta_2$ from S_1 and S_2, respectively.

Step 8. If θ_1 and θ_2 are consistent, then S is unsatisfiable.

In the above procedure, S is decomposed into S_1 and S_2. If S_1 and S_2 can be proved unsatisfiable under a consistent substitution for parameters occurring in C_1^* and C_2^*, we can conclude that S is unsatisfiable. Since we want to decompose S into as simple S_1 and S_2 as possible, in Step 1 of the above procedure we usually choose a long clause in S to split. Of course, if either S_1 or S_2 still contains long clauses, it can be decomposed again. Furthermore, in Step 7, there may be many derivations of the empty clause from S_1 and S_2. We should try to derive empty clauses $\square | \theta_1$ and $\square | \theta_2$ from S_1 and S_2 respectively such that θ_1 and θ_2 are consistent. In Step 7, many refinemnets of V-resolution, such as hyper V-resolution, lock V-resolution, set-of-support V-resolution linear V-resolution and unit V-resolution [4] can be used.

We should note here that sometimes either S_1 or S_2 may not be proved unsatisfiable, even though S is unsatisfiable. In this case, S should be enlarged by introducing additional copies of the splitting clause C into S. For detail, see [3] or Chapter 9 of [4]. We now give an example to illustrate the decomposition procedure.

Example 3. Consider the following theorem: In an associative system with left and right solutions, there is a right identity element. Let $P(x,y,z)$ denote $x \circ y = z$. Then, the negation of the above theorem can be represented by the following set S of clauses, where x,y,z,u,v and w are variables:

$$S = \begin{cases} P(g(x,y),x,y) | \varepsilon & (1) \\ P(x,h(x,y),y) | \varepsilon & (2) \\ {\sim}P(x,y,u) \vee {\sim}P(y,z,v) \vee {\sim}P(x,v,w) \vee P(u,z,w) | \varepsilon & (3) \\ {\sim}P(k(x),x,k(x)) | \varepsilon & (4) \end{cases}$$

Suppose we choose clause (3) to split. That is,
$C = {\sim}P(x,y,u) \vee {\sim}P(y,z,v) \vee {\sim}P(x,v,w) \vee P(u,z,w)$.
Let $C_1 = {\sim}P(x,y,u) \vee {\sim}P(y,z,v)$ and $C_2 = {\sim}P(x,v,w) \vee P(u,z,w)$.
Note that $C = C_1 \vee C_2$. The variables which occur in both C_1 and C_2 are x,z,u and v. Suppose we use integers to represent parameters. We use parameters 1, 2, 3, and 4 for x,z,u and v, respectively. Thus,
$C_1^* = {\sim}P(1,y,3) \vee {\sim}P(y,2,4)$ and $C_2^* = {\sim}P(1,4,w) \vee P(3,2,w)$.

S is then decomposed into S_1 and S_2 as follows:

$$S_1 = \begin{cases} P(g(x,y),x,y) \mid \varepsilon \\ P(x,h(x,y),y) \mid \varepsilon \\ {\sim}P(1,y,3) \vee {\sim}P(y,2,4) \mid \varepsilon \\ {\sim}P(k(x),x,k(x)) \mid \varepsilon \end{cases}$$

$$S_2 = \begin{cases} P(g(x,y),x,y) \mid \varepsilon \\ P(x,h(x,y),y) \mid \varepsilon \\ {\sim}P(1,4,w) \vee P(3,2,w) \mid \varepsilon \\ {\sim}P(k(x),x,k(x)) \mid \varepsilon \end{cases}$$

From S_1, we can derive an empty clause $\square \mid \{ g(5,6)/1,\ h(5,7)/2,\ 6/3,\ 7/4 \}$ as shown in Fig. 2. From S_2, we can also derive an empty clause $\square \mid \{ g(9,k(8))/1,\ 8/2, k(8)/3,\ 9/4,\ k(8)/10 \}$ as shown in Fig. 3. Since $\{ g(5,6)/1,\ h(5,7)/2,\ 6/3,\ 7/4 \}$ and $\{ g(9,k(8))/1,\ 8/2, k(8)/3,\ 9/4,\ k(8)/10 \}$ are consistent, we conclude that S is unsatisfiable.

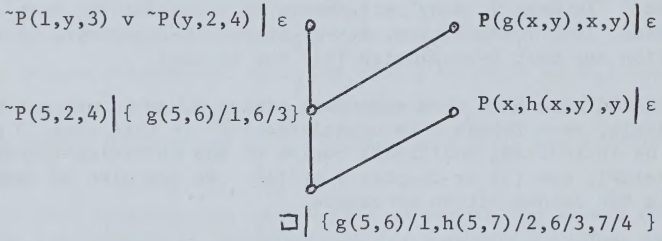

~P(1,y,3) v ~P(y,2,4) | ε
P(g(x,y),x,y) | ε
~P(5,2,4) | { g(5,6)/1,6/3}
P(x,h(x,y),y) | ε
□ | {g(5,6)/1,h(5,7)/2,6/3,7/4 }

Figure 2

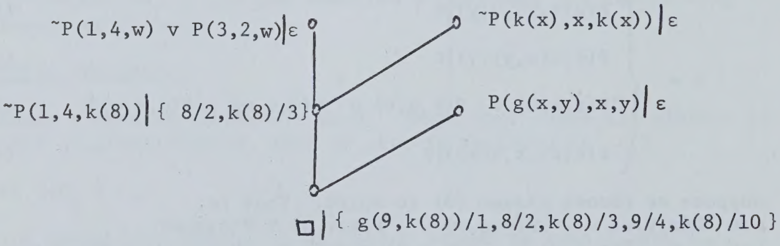

~P(1,4,w) v P(3,2,w) | ε
~P(k(x),x,k(x)) | ε
~P(1,4,k(8)) | { 8/2,k(8)/3}
P(g(x,y),x,y) | ε
□ | { g(9,k(8))/1,8/2,k(8)/3,9/4,k(8)/10 }

Figure 3

4. Experiments

A LISP computer program was written to test the decomposition method discussed in the last section. For simplicity, the program used the breadth-first technique to generate V-resolvents. That is, given a set S of clauses, the program computed a sequence of sets of clauses, $R(S)$, $R^1(S)$,..., until empty clauses were computed, where

$$R^0(S) = S,$$

$$R^n(S) = \{\text{V-resolvents from } R^0(S) \cup \ldots \cup R^{n-1}(S)\} \quad n=1,2,\ldots$$

We used Example 3 given in the last section as our test example. The program was compiled and run on the PDP-10 time-sharing system. Let S, S_1 and S_2 be the sets of clauses given in Example 3. We let the program run on S, S_1 and S_2. For S, it took the program 13.27 seconds to find the empty clauses. However, for S_1 and S_2, it took respectively 0.65 and 0.93 seconds to find the empty clauses. Thus, if S was decomposed into S_1 and S_2, it took the program a total of 1.58 seconds to prove the theorem. Compared with the case where S is not split, this is a saving of about 88% in time. Of course, in the case that S is decomposed into S_1 and S_2, after empty clauses are obtained from S_1 and S_2, we have to check the consistency of substitutions for parameters. Still, applying the decomposition method may yield a great savings both in time and in space.

5. Concluding Remarks

Proving a theorem by considering separately each of all its special cases is often used in mathematics. In this paper, we give a method to decompose a set of clauses by spitting long clauses in the set. Experimental results show that proving a theorem by using the decomposition method may yield a great savings.

References

1. Bledsoe, W. W. (1971): Splitting and reduction heuristics in automatic theorem proving, Artificial Intelligence, Vol. 2, No. 1, Spring, 1971 pp. 57-78.

2. Chang, C. L. (1971): Theorem proving by generation of pseudo-semantic trees, Division of Computer Research and Technology, National Institutes of Health, Bethesda, Maryland, 1971.

3. Chang, C. L. (1972): Theorem proving with variable-constrained resolution, Information Sciences 4, 217-231.

4. Chang, C. L., and Lee, R. C. T. (1972): Symbolic Logic and Mechanical Theorem Proving, to be published by Academic Press in Jan.,1973.

5. Ernst, G. (1971): The utility of independent subgoals in theorem proving, Information and Control, Vol. 18, No. 3, April 1971, pp. 237-252.

6. Prawitz, D. (1960): An improved proof procedure, Theoria, Vol. 33, 1967, pp. 246-254.

7. Prawitz, D. (1969): Advances and problems in mechanical proof procedure Machine Intelligence 4 (edited by B. Meltzer and D. Michie), American Elsevier, New York, 1969, 1969, pp.59-71.

8. Robinson, J. A. (1965): A machine-oriented logic based on the resolution principle, Journal of the ACM, Vol. 12, No. 1, Jan 1965, pp. 23-41.

9. Slagle, J. R. and Koniver, D. (1971): Finding resolution proofs and using duplicate goals in AND/OR trees, Information Sciences, Vol. 4, No. 4, Oct. 1971, pp. 315-342.

10. Wos, L., Robinson, G. A., Carson, D. F. and Shalla, L. (1967): The concept of demodulation in theorem proving, Journal of the ACM, Vol. 15, No. 4, Oct. 1967, pp. 698-709.

STABILITY REGIONS OF LARGE-SCALE SYSTEMS

STEIN WEISSENBERGER
University of Santa Clara, Santa Clara, California 95053

ABSTRACT

Using an aggregated comparison system, methods are developed for estimating regions of asymptotic stability for large-scale systems composed of interconnected, exponentially stable subsystems; in general some of these subsystems are not globally stable. Three different forms of Liapunov functions are shown to be suitable for describing the stability behavior of the aggregated system, but one form in particular (heretofore not used in this context) is found to be the most natural. Using this function, some general results are obtained.

I. INTRODUCTION

This paper is concerned with the problem of describing non-global stability behavior of dynamic systems of high dimension. A particular objective is the estimation of the region of asymptotic stability (RAS), an important problem which has long presented computational difficulties for large (i.e., greater than two or three) dimensional systems. In attacking this problem, we will make use of the aggregation approach to large-scale system analysis introduced originally by Bailey [1] and continued with variations more recently by others [2-7]. The ideas which unify all this work are: (1) The description of the system as a collection of interconnected subsystems, each characterized by a strong stability property, typically exponential stability, and (2) system stability analysis based on the properties of (given) subsystem Liapunov functions and the description of interconnection functions through "conical" bounds of the form $|g(x)| \leq \xi\ |x|$. In some cases, a dynamic comparison system (whose states are subsystem Liapunov functions) is constructed, while in others a scalar Liapunov function is constructed from subsystem functions, without the explicit identification of a comparison system. In the present work, use is made of both a comparison system and an overall system scalar Liapunov function.

The problem of estimating the RAS has been considered previously in the aggregation literature only by Thompson and Koenig [7]; their technique, when applied to this problem (which plays a minor role in their work) appears to lead to results which are rather more conservative than ours; their results are also, in contrast to ours, inherently numerical in character.

Among the problems which are considered in this paper is the following: To find conditions under which a system composed of interconnected subsystems is asymptotically stable for $x(0) \varepsilon R$, where $x^T = [x_1^T, x_2^T, \ldots x_s^T]$, $R = R_1 \times R_2 \times \ldots \times R_s$, x_i is the state of the ith subsystem, and R_i is the RAS of the ith subsystem. Similar problems have been treated in the work of Michel [8] and Weiss and Infante [9], which use Liapunov-like methods without relying on aggregation techniques. Although they discuss stability properties valid over sub-sets of the state space, their concern is not with asymptotic stability, but rather boundedness properties such as stability under perturbing forces and finite-time stability.

The present paper begins in Section II with definitions, a basic description of the system, and a statement of the fundamental comparison results, modified for the study of non-global stability properties. The system is assumed to consist of interconnected (at least locally) exponentially stable subsystems; in general, some subsystems are not globally stable. Section III presents theorems concerning the estimation of regions of asymptotic stability and regions of invariance, and provides conditions under which the two regions are identical, using a Liapunov function for the comparison system. Three specific Liapunov functions, V_1, V_2, V_3, are introduced, and these are employed in turn for the calculation of RAS estimates in Sections IV, V, and VI. The function V_3 turns out to be the most natural and useful one, and some of the major results of the paper are obtained with it in Section VII. An example is given in Section VIII, and Section IX states conclusions. For reasons of brevity, proofs are frequently either omitted or sketched; complete proofs may be found elsewhere [10]. Reference [10] also contains other omitted material, including a discussion of the estimation of the convergence rate.

II. PRELIMINARIES

Consider the system*

$$(S) \qquad \dot{x}_i = f_i(x_i) + \sum_{j=1}^{s} g_{ij}(x_j,t) \quad i = 1,2, \ldots, s \tag{2-1}$$

which is composed of subsystems

$$(S_i) \qquad \dot{x}_i = f_i(x_i) \qquad i = 1,2, \ldots, s \tag{2-2}$$

interconnected through functions $g_{ij}(x_j,t)$. The subsystem function f_i and the state x_i of each subsystem is n_i dimensional; the state x of (S) is defined as

$$x^T \triangleq [x_1^T, x_2^T, \ldots, x_s^T]$$

and is of dimension $\sum_{i=1}^{s} n_i$. The state space of each subsystem (S_i) is represented by E_i, and of the system (S) by E, with

$$E = E_1 \times E_2 \times \ldots \times E_s.$$

Existence and uniqueness of solutions of (2-1) and (2-2) are assumed throughout all regions of interest. It is assumed that $\underline{x} = 0$ is an equilibrium solution of (2-1), i.e., that

$$f_i(0) = 0, \quad i = 1,2, \ldots, s$$
$$g_{ij}(0,t) = 0, \quad i,j = 1,2, \ldots, s, t \varepsilon T,$$

where

$$T = \{t : t \geq 0\}$$

Throughout the paper we refer to the following

Definition. The region R' is the region of asymptotic stability (RAS) of the system

$$\dot{x} = h(x,t), \; h(0,t) = 0, \; t \varepsilon T,$$

if and only if

i) $x = 0$ is stable in the sense of Liapunov,

and ii) $\lim_{t\to\infty} x\,[x(t_o), t_o, t] = 0$ for $t_o \varepsilon T$ if and only if $x(t_o) \varepsilon R'$.

A system will be said to be exponentially stable in a region R' if, in addition, solutions satisfy a bound $|x[x(t_o), t_o, t]| \leq \beta |x(t_o)| e^{-\alpha(t-t_o)}$ for some numbers $\alpha > 0$, $\beta > 0$, for all $x(t_o) \varepsilon R'$, $t_o \varepsilon T$.

It is assumed that the equilibrium of each uncoupled subsystem (S_i) is (locally) exponentially stable; that each subsystem (S_i), $i = 1,2, \ldots \ell$ is globally exponentially stable, while each remaining subsystem (S_i), $i = \ell + 1, \ell + 2, \ldots, s$ is exponentially stable only in a region R_i' which does not constitute its state space E_i. Thus we have ℓ globally stable

---

[*] With some obvious exceptions, lower case Roman letters indicate vectors, lower case Greek letters scalars, upper case Roman letters matrices or sets, and upper case Greek letters sets.

subsystems and s-ℓ non-globally stable subsystems. Our problem is to estimate, for the system (S), the extent of the region of asymptotic stability (RAS) of the solution x = 0, and in particular to compare the estimate of this region to estimates of the RAS of (S) with zero interconnections,

$$R' = E_1 \times E_s \times \dots \times E_\ell \times R'_{\ell+1} \times R'_{\ell+2} \times \dots \times R'_s \qquad (2\text{-}3)$$

The first stage of the attack on this problem is to construct a comparison system based on subsystem Liapunov functions. This process follows closely that applied to global stability problems in [4], except that certain restrictions are imposed on the allowable values of the state. Since the equilibrium of each subsystem is exponentially stable, there exist non-negative, continuous scalar functions $v_i(x_i)$, $v_i(x_i) > 0$, $x_i \neq 0$, and $v_i(0) = 0$, with the following properties:

$$\eta_{i1}|x_i| \leq v_i \leq \eta_{i2}|x_i| \qquad (2\text{-}4)$$

$$\dot{v}_{iS_i} \leq \eta_{i3}|x_i| \qquad (2\text{-}5)$$

$$|\nabla v_i| \leq \eta_{i4} \qquad (2\text{-}6)$$

where $\dot{v}_{iS_i}$ is the derivative of v_i along trajectories of the subsystem S_i, for the following values of x_i:

$$x_i \in E_i, \quad i = 1,2, \dots, \ell \qquad (2\text{-}7)$$

$$x_i \in R_i, \quad i = \ell+1, \ell+2, \dots, s, \qquad (2\text{-}8)$$

where the regions R_i are given by

$$R_i \triangleq \{x_i: \; v_i(x_i) < v_{oi}\}, \quad i = \ell+1, \ell+2, \dots, s. \qquad (2\text{-}9)$$

The functions v_i are thus subsystem Liapunov functions, while the regions R_i are estimates of the regions R'_i obtained by Liapunov functions, where $R_i \subseteq R'_i$. There are a number of effective procedures for determining Liapunov functions for globally stable (low-order) subsystems, e.g., [11,12], as well as for calculating estimates R_i of the RAS, e.g., [13-17].*

The region R,

$$R \triangleq E_1 \times E_2 \times \dots \times E_\ell \times R_{\ell+1} \times R_{\ell+2} \times \dots \times R_s \qquad (2\text{-}10)$$

is an estimate of the RAS of (S) with all interconnections zero.

We now assume that there exist numbers $\xi_{ij} > 0$, such that,

$$\text{for} \quad x \in R, \; |g_{ij}(x_j,t)| \leq \xi_{ij} \; |x_j| \;, \; i,j = 1,2, \dots, s, \; t \in T. \qquad (2\text{-}11)$$

The numbers ξ_{ij} will be required subsequently to satisfy certain conditions. The numbers v_{oi}, i = ℓ+1, ..., s, in the specification of R arise solely from calculation of subsystem RAS, R_i.

Now we make use of the inequalities (2-11) and the subsystem Liapunov functions with properties (2-4)-(2-6) to form on the trajectories of (S) the comparison system of inequalities,

$$(C) \qquad \dot{v} \leqq A\, v^{**}, \; t \in T, \; x \in R, \qquad (2\text{-}12)$$

$$\text{where} \quad v^T = [v_1, v_2, \dots, v_s],$$

---

[*] Certain (obvious) modifications of these methods are required to obtain the properties(2-4,5,6),e.g., here one takes the square-root of a quadratic as Liapunov function and one must make a simple transformation of a subsystem equation in order to obtain the exponential stability property (2-5).

[**] Notation:
$a<b$ iff $a_i<b_i \, \forall i$; $a \leq b$ iff $a_i \leq b_i \, \forall i$ and $a_i \neq b_i$ for some i; $a \leqq b$ iff $a_i \leq b_i \, \forall i$.

and
$$a_{ij} = -\delta_{ij}\,\eta_{i2}^{-1}\,\eta_{i3} + \xi_{ij}\,\eta_{j1}^{-1}\,\eta_{i4}, \tag{2-13}$$
where δ_{ij} is the Kronecker delta.

In making use of comparison inequality (2-12) to describe the motion of (2-1), we encounter the comparison equalities,
$$(C_1)\ \dot{w} = Aw,\ w(0) = v(0)^{*} \tag{2-14}$$

We immediately note that the matrix A has non-negative off-diagonal elements and negative diagonal elements:
$$a_{ij} \geq 0,\ i \neq j;\ a_{ij} < 0,\ i = j;\ i,\ j = 1,2,\ \ldots,\ s, \tag{2-15}$$
i.e., A is a Metzler matrix [18]. This property implies the important comparison property [19],
$$w(t) \geq v(t) \geq 0,\ \forall t \geq 0,\ x \epsilon R. \tag{2-16}$$
(The non-negativity of v(t) follows trivially from the definition of v.)

We now make the additional requirement, which will hold subsequently throughout the paper, that the matrix A is Hurwitz (i.e., A is a stable Metzler matrix),
$$\mathrm{Re}\,[\lambda_i(A)] < 0,\ i = 1,2,\ \ldots,\ s. \tag{2-17}$$
This condition allows us to infer stability properties of the original system (2-1): Since (2-14) is asymptotically stable, making use of (2-16) leads us to conclude that
$$\lim_{t\to\infty} v(t) = 0 \tag{2-18}$$
and that in fact (2-1) is asymptotically stable, all of course under the provision that $x \epsilon R$. Precise and detailed implications of this latter provision form the main substance of the remainder of the paper.

There is a simple necessary and sufficient condition for a Metzler matrix A to be Hurwitz, the Hicks condition [20]: the minors of A alternate in sign,
$$a_{11} < 0,\quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0,\ \ldots,\ (-1)^{s} \det A > 0^{**} \tag{2-19}$$

III. ESTIMATION OF STABILITY REGIONS

Consider a non-negative, continuous scalar function V(v), with $V(v) > 0$, $v \neq 0$, and $V(0) = 0$. Define the region $\tilde{R}$ as
$$\tilde{R} \triangleq \{x:\ V\,[v(x)] < \gamma\},\ \gamma = \underset{v \epsilon R_b}{\mathrm{minimum}}\ V(v)\ , \tag{3-1}$$
where $R_b = \bigcup_{i=\ell+1}^{s} \{v:\ v_i = v_{oi};\ v_j \geq 0,\ j=1,2,\ldots,\ell;\ v_j \leq v_{oj},\ j \neq i,\ j=\ell+1,\ldots,s\}$

We will be concerned with the sign properties of the derivative of V(v) on trajectories of (2-14),
$$\dot{V}_{C_1} = \nabla V^{T} Av. \tag{3-2}$$
We assume in Theorems 1-3 that (2-19) holds.

Theorem 1. If $v(0)\ \epsilon\ \tilde{R}$ and $V_{C_1}(v) \leq 0$ for $v \epsilon R$, then $v(t)\ \epsilon \Phi$, $t \geq 0$, where
$$\Phi = \bigcup_{w \epsilon \tilde{\tilde{R}}} \Phi_w$$
where $\Phi_w = \{v:\ v < w\}$ and $\tilde{\tilde{R}}$ is the closure of $\tilde{R}$,
i.e., a trajectory initiating in $\tilde{R}$ never leaves $\Phi \subseteq R$.

The proof follows immediately from (2-16) and the assumption $V(v) \leq 0$ for $v \epsilon \tilde{R}$. Furthermore, we readily establish that $\lim_{t\to\infty} v(t) = 0$ to give the next

---

[*] Since the comparison systems (2-12, 14) are time invariant and valid for all $t \epsilon T$, we subsequently restrict our attention to $t_o = 0$ without loss of generality.

[**] A useful collection of properties of such matrices is given in [21].

result:

Theorem 2. The region $\tilde{R}$ is an estimate of the RAS of (2-1), $\tilde{R} \subseteq \Phi \subseteq R \subseteq R'$, if

$$\dot{V}_{C_1}(v) \leq 0,\ v \neq 0,\ \dot{V}_{C_1}(0) = 0, \text{ for } v \epsilon R.$$

The next result gives a sufficient condition for $\tilde{R} = \Phi$, i.e., that the estimate $\tilde{R}$ of the RAS is an invariant region:

Theorem 3. If $\nabla V(v) \geq 0$, for $v > 0$, $v \epsilon R$, $\tilde{R} = \Phi$.

The proof follows directly from the definition of Φ and the sign properties of ∇V, assuming that V is continuously differentiable. It is also instructive to show that R is an invariant region when $\nabla V \geq 0$, $v \neq 0$, by an argument which does not require continuity of ∇V: If $\nabla \bar{V} \geq 0$, then we can write

$$\dot{V}_C = \nabla V^T \dot{v}_C \leqq \nabla V^T Av = \dot{V}_{C_1} \tag{3-4}$$

Hence if V is a Liapunov function for (C_1) it is also one for the inequality system (C) (and hence the original system (2-1)).

We will consider the following candidates for Liapunov functions for (C_1):

$$V_1 = v^T Hv,\ H > 0^* \tag{3-5}$$

$$V_2 = \sum_{i=1}^{s} r_i v_i,\ r_i > 0 \quad i = 1,2, \ldots, s \tag{3-6}$$

$$V_3 = \underset{i=1,2,\ldots,s}{\text{maximum}} (v_i/p_i),\ p_i > 0 \quad i = 1,2, \ldots, s \tag{3-7}$$

The functions V_3 $(p_i = 1)$ and V_2 were introduced by Rosenbrock [22] in a somewhat different context. The function V_2 has been further studied by Mitra and So [23] and also applied to the interconnected-system global stability problem (3-7). We briefly review the properties of V_1 and V_2 and subsequently concentrate on V_3.

IV. FUNCTION V_1

To calculate $\tilde{R}$ for the quadratic form V_1 we first form $\dot{V}_{1C_1} = - x^T Qx$, and require it to be at least negative semi-definite:

$$- Q = HA + A^T H,\ Q \geq 0. \tag{4-1}$$

For each Q > 0 there is a unique solution H > 0 to (4-1). Suppose that $V_1 = x^T Hx$ has been so constructed; recalling (3-1) we can perform the indicated minimization to find that

$$\gamma = \underset{1 \leq i \leq s}{\text{minimum}} \{(\bar{h}_{ii})^{-1} v_{oi}^2\} = (\det H) \underset{1 \leq i \leq s}{\text{minimum}} \{\frac{v_{oi}^2}{\det M_i}\} \tag{4-2}$$

where $\bar{h}_{ii}$ is the iith element of H^{-1} and M_i is ith principal minor of H.

V. FUNCTION V_2

Form $\dot{V}_{2C_1} = r^T Av \leq 0$; we can insure $\dot{V}_2 < 0$ by choosing r > 0 such that

$$r^T A < 0. \tag{5-1}$$

Such an r exists [21] if A satisfies the Hicks conditions. In this case we can always construct r by choosing any $q^T > 0$ and setting [23] $r^T A = - q^T < 0$, and solving for

$$r^T = -q^T A^{-1}. \tag{5-2}$$

Since the inverse matrix A^{-1} exists and is non-positive if A satisfies the Hicks conditions, it follows that every column of A^{-1} has at least one negative element, guaranteeing r > 0. Note that since $\nabla V = r > 0$, $\tilde{R} = \Phi$. Now, applying (3-1) we find the region $\tilde{R}$ to be determined from

$$\gamma = \underset{1 \leq i \leq s}{\text{minimum}}\ r_i v_{oi} \tag{5-3}$$

---

[*] H > 0 means that H is positive definite.

VI. FUNCTION V_3

From (3-7), the region $\tilde{R}$ is given by

$$\tilde{R} = \{v: \; V(v) < \gamma\} = \{v: \; v < \hat{v} = \gamma p\}. \tag{6-1}$$

It is easy to show that

$$\dot{V}_3 \le \underset{1 \le i \le s}{\text{supremum}} \; p_i^{-1} \sum_{j=1}^{s} a_{ij} v_j, \; v \varepsilon \tilde{R}, \tag{6-2}$$

and hence that

$$\dot{V}_3 \le 0 \tag{6-3}$$

if

$$Ap \le 0. \tag{6-4}$$

(Because A is non-singular and $p > 0$ the possibility $Ap = 0$ is excluded.) For the strict inequlity in (6-3), $\dot{V}_3 < 0$, we obtain the stronger condition

$$Ap < 0. \tag{6-5}$$

Solutions $p > 0$ to (6-5) are found, similarly to Section V, by making use of the fact that a non-positive inverse A^{-1} exists if A is a stable Metzler matrix and now that every row of A^{-1} has at least one negative element; then for any $q > 0$ we have

$$p = - A^{-1} q > 0. \tag{6-6}$$

According to Theorems 2 and 3 then, if $p > 0$ satisfies (6-5), $\tilde{R}_3$ is an estimate of the RAS, with $\tilde{R}_3 = \Phi$.

Assuming now that a solution $p > 0$ to (6-4) has been found, e.g., by (6-6), we compute γ by (3-1) to be

$$\gamma = \underset{1 \le i \le s}{\text{minimum}} \; (v_{oi}/p_i). \tag{6-7}$$

Note from (6-2) the important property that unlike $\tilde{R}_1$ and $\tilde{R}_2$, the region $\tilde{R}_3$ is uncoupled in the subsystem functions v_i, an enormous simplification in interpretation of $\tilde{R}$. Also, we see that since $\tilde{R}_3$ is of the same form as R, there is the possibility of achieving $\tilde{R} = R$, clearly impossible with either V_1 or V_2. Consequently, we proceed to concentrate primarily on the function V_3.

VII. MAXIMUM-VOLUME ESTIMATES

The volume ν of $\tilde{R}$ is given by

$$\nu = 2^s \gamma^s \prod_{i=1}^{s} p_i, \; \gamma = \underset{1 \le i \le s}{\text{minimum}} \; (v_{oi}/p_i). \tag{7-1}$$

Hence the problem of determining the "best" estimate of the RAS in the sense of the estimate containing the largest volume, ν^*, is easily formulated as the following maximization problem:

$$\text{maximize } \nu(p) \text{ subject to the constraints } Ap \le 0, \; p > 0, |p| = 1. \tag{7-2}$$

The optimum estimate, R^*, is then given by $R^* = \{v: \; v < \gamma^* p^*\}$. It is assumed in (7-2) that ν^* is finite, i.e., that $v_{oi} \neq \infty \; \forall i$. The general solution to (7-2) for $s = 2$ may be calculated and found to yield the explicit region estimate

$$\tilde{R} = \{v: \; v_1 < v_1^*, \; v_2 < v_2^*\}, \tag{7-3}$$

where

$$v_1^* = \min \{v_{o1}, \max_{i=1,2} (- v_{o2} \, a_{i2}/a_{i1})\}$$

$$v_2^* = \min \{v_{o2}, \max_{i=1,2} (- v_{o1} \, a_{i1}/a_{i2})\}$$

The optimal estimate (7-3) has a further interesting (and unusual) property: It contains all other estimates obtained with the form V_3, this providing in itself all possible information about the extent of the RAS obtainable with V_3: we have, for $s = 2$,

$$\tilde{R}|_p \subseteq \tilde{R}|_{p^*} \tag{7-4}$$

where p^* is the solution to (7-2) and p is any positive vector satisfying (6-4). The situation for $s > 2$ is more complicated; explicit region estimates are more difficult to obtain, although the problem (7-2) remains a straightforward one. The property (7-4) does not hold for all maximum-volume estimates for $s > 2$, though it can be shown by example to hold for some.

Of considerable interest are conditions which guarantee that the entire region R is a RAS. Hence, we pose the following Problem: Under what conditions will the introduction of the interconnections of (2-1) not decrease the stability region $\tilde{R}$ of the system (2-2) from its unconnected extent R? An answer is supplied by the following two theorems:

Theorem 4. Suppose that A is a stable Metzler matrix. For $\ell = 0$ and $v_o > 0$, a sufficient condition that

$$\tilde{R} = R = \{x: \ v(x) < v_o\} \tag{7-5}$$

is

$$Av_o \leq 0. \tag{7-6}$$

Note that since A is a Metzler matrix, satisfaction of a strict inequality in (7-6) guarantees that A is Hurwitz, thus eliminating the necessity of a separate test of the Hicks conditions (2-19). (The condition $Ap < 0, p > 0$ is both necessary and sufficient for A to be Hurwitz when A is Metzler [21].)

Proof: From (6-7) it is clearly sufficient to take $p = v_o$, obtaining

$$\tilde{R} = \{x: \ V_3[v(x)] < 1\} = \{x: \ v(x) < v_o\} = R.$$

We use this result to approach the general case, $\ell \geq 0$:

Theorem 5. Suppose that A is a stable Metzler matrix. A sufficient condition that

$$\tilde{R} = R = E_1 \times \ldots \times E_\ell \times R_{\ell+1} \times \ldots \times R_s$$

is that

$$A^{22} v_o^2 \leq 0 \tag{7-7}$$

and

$$a_{ij} = 0, \ i = \ell+1, \ell+2, \ldots, s, \ j = 1,2, \ldots, \ell \tag{7-8}$$

where

$$A^{22}_{i-\ell,j-\ell} = A_{ij}, \ v^2_{oi-\ell} = v_{oi}, \ i,j = \ell+1, \ell+2, \ldots, s. \tag{7-9}$$

The proof follows from the construction according to Theorem 4 of estimates $\tilde{R}$ which are identical to a bounded region R_γ which can be made as close as desired to the region R by choosing the number $\gamma > 0$ sufficiently large.

Observe that satisfaction of Theorem 5 requires that globally stable subsystems not influence finitely stable subsystems.

VIII. EXAMPLE

Consider the following system:

$$\begin{aligned}
\dot{x}_1 &= - x_1 + g_{12}(x_2) + g_{13}(x_3) \\
\dot{x}_2 &= - x_2 + x_2^3 + g_{21}(x_1) + g_{23}(x_3) \\
\dot{x}_3 &= - x_3 + x_3^3 + g_{31}(x_1) + g_{32}(x_2)
\end{aligned}$$

Take $v_i = |x_i|$, $i = 1,2,3$, giving $\eta_{i1} = \eta_{i2} = \eta_{i4} = 1$, $i = 1,2,3$,

$$\eta_{13} = 1, \ \eta_{23} = 1 - v_{o2}^2 > 0, \ \eta_{33} = 1 - v_{o3}^2 > 0 \ ,$$

with

$$\begin{aligned}
R_1 &= E_1 = \{x_1\}, \\
R_2 &= R_2^+ = \{x_2: \ |x_2| < v_{o2}\}, \\
R_3 &= R_3^+ = \{x_3: \ |x_3| < v_{o3}\}
\end{aligned}$$

Note the accuracy of the RAS representation by R_2 and R_3.

Then

$$A = \begin{bmatrix} -1 & \xi_{12} & \xi_{13} \\ \xi_{21} & -(1-v_{o2}^2) & \xi_{23} \\ \xi_{31} & \xi_{32} & -(1-v_{o3}^2) \end{bmatrix}$$

Using Theorem 5, in order to have $\tilde{R} = R = E_1 \times R_2 \times R_3 = R'$, we require that

$$\xi_{21} = \xi_{31} = 0$$

and, in satisfying (7-7), that

$$\xi_{23} < (1-v_{o2}^2)\, v_{o2}/v_{o3}, \text{ and } \xi_{32} < (1-v_{o3}^2)\, v_{o3}/v_{o2}.$$

In addition we require that A is Hurwitz. This can be accomplished by requiring also that

$$\xi_{12} + \xi_{13} < 1.$$

IX. CONCLUSIONS

A method has been developed for computing estimates of the region of asymptotic stability of a collection of interconnected subsystems. Best results were obtained using a comparison-system Liapunov function of a form originally introduced by Rosenbrock. The estimates produced with this function are "uncoupled" in terms of subsystem Liapunov functions, greatly facilitating the interpretation of RAS estimates. The problem of computing maximum-volume estimates is readily defined for this function and is explicitly solved for s = 2. The special properties of this function also permit conditions to be found under which the coupled and uncoupled system RAS estimates are identical, i.e., conditions under which the introduction of coupling between subsystems does not reduce the estimate of the RAS of any subsystem.

ACKNOWLEDGMENT

This research was supported under contract SCC-306 from the NASA-Ames Research Center.

REFERENCES

[1] Bailey, F. N., "The Application of Lyapunov's Second Method to Interconnected Systems", J. SIAM Control, Vol. 3, pp. 443-462, 1966.

[2] Piontkovskiĭ, A. A., and L. D. Rutkovskaya, "Investigation of Certain Stability Theory Problems by the Vector Liapunov Function Method", Automation and Remote Control, Vol. 28, pp. 1422-1429, Oct. 1967.

[3] Thompson, W. E., "Exponential Stability of Interconnected Systems", IEEE Trans. Automat. Contr., Vol. AC-15, pp. 504-506, Aug. 1970.

[4] Šiljak, D. D., "Stability of Large-Scale Systems under Structural Perturbations", Ninth Allerton Conference on Circuit and System Theory, 1971, to appear in IEEE Trans. on Systems, Man, and Cybernetics, Nov. 1972.

[5] Michel, A. N., and D. W. Porter, "Stability Analysis of Composite Systems", IEEE Trans. Automat. Contr., Vol. AC-17, pp. 222-226, April 1972.

[6] Araki, M., and B. Kondo, "Stability and Transient Behavior of Composite Nonlinear Systems", IEEE Trans. Automat. Contr., Vol. AC-17, pp. 537-541, Aug. 1972.

[7] Thompson, W. E., and H. E. Koenig, "Stability of a Class of Interconnected Systems", *Int. J. Control*, Vol. 15, *4*, pp. 751-763, 1972.

[8] Michel, A. N., "Stability, Transient Behavior and Trajectory Bounds of Interconnected Systems", *Int. J. Control*, Vol. 11, *4*, pp. 703-715,1970.

[9] Weiss, L., and Infante, E. F., "Finite Time Stability under Perturbing Forces and on Product Spaces", *IEEE Trans. Automat. Contr.*, Vol. AC-12, pp. 54-59, Feb. 1967.

[10] Weissenberger, S., *Stability Regions of Large-Scale Systems*, Report SCC-306, University of Santa Clara, Oct. 1972.

[11] Ingwerson, D. R., "A Modified Lyapunov Method for Nonlinear Stability Analysis", *IRE Trans. Automat. Contr.*, Vol. AC-6, *2*, pp. 199-210, May 1961.

[12] Kalman, R. E., "Lyapunov Functions for the Problem of Lur'e in Automatic Control", *Proceedings of the National Academy of Sciences*, Vol. 49, *2*, pp. 201-205, Feb. 1963.

[13] Margolis, S. G., and Vogt, W. G., "Control Engineering Applications of V.I. Zubov's Construction Procedure for Lyapunov Functions", *IEEE Trans. Automat. Contr.*, Vol. AC-6, pp. 199-210, May 1961.

[14] Weissenberger, S., "Stability-Boundary Approximations for Relay-Control Systems via a Steepest-Ascent Construction of Liapunov Functions", Trans. ASME, *J. Basic Engr.*, Sec. D, Vol. 88, pp. 419-428, June 1966.

[15] Nelson, W. E., "Nonlinear Control Systems Analysis by Liapunov's Direct Method", M.S. Thesis, Dept. of Elec. Engr., University of Santa Clara, 1967.

[16] Walker, J. A., and McClamroch, N. H., "Finite Regions of Attraction for the Problem of Lur'e", *Int. J. Control*, Vol. 6, *4*, pp. 331-336, 1967.

[17] Weissenberger, S., "Application of Results from the Absolute Stability Problem to the Computation of Finite Stability Domains", *IEEE Trans. Automat. Contr.* (correspondence), Vol. AC-13, pp. 124-125, Feb. 1968.

[18] Arrow, K. J., "Price-Quantity Adjustments in Multiple Markets with Rising Demands", in *Mathematical Methods in the Social Sciences*, (1959), Ed. by K. J. Arrow, et al., Stanford U. Press, Stanford, 1966, pp. 3-15.

[19] Bellman, R., "Vector Liapunov Functions", *J. SIAM Control*, Vol. 1, *1*, pp. 32-34, 1962.

[20] Quirk, J., and Saposnik, R., *Introduction to General Equilibrium Theory and Welfare Economics*, McGraw-Hill, N.Y., 1968.

[21] Fiedler, M., and Ptak, P., "On Matrices with Non-Positive Off-Diagonal Elements and Positive Principal Minors", *Czech. Math. J.*, Vol. 12, *3*, pp. 382-400, 1962.

[22] Rosenbrock, H. H., "A Method of Investigating Stability", 1963 IFAC Proceedings, Basle.

[23] Mitra, D., and So, H. C., "Linear Inequalities and P Matrices, with Applications to Stability of Nonlinear Systems", Fifth Asilomar Conference on Circuits and Systems, pp. 179-188, 1971.

PARTIAL IDENTIFICATION OF LARGE-SCALE SYSTEMS

R. P. Guidorzi and G. Marro
Istituto di Automatica
University of Bologna
40136 Bologna, Italy

ABSTRACT

The identification of large-scale systems from input/output data is approached in this paper by separately identifying a set of subsystems, each completely describing the input/output dynamical behavior between the input and a limited group of output components.

The order of these subsystems is usually much smaller than the whole system one and allows therefore the use of conventional techniques for their structural and parametric identification.

I. INTRODUCTION

The complete identification of a dynamical model for a large-scale system (LSS) can prove to be, in practice, a really huge problem. In fact the use of well-proven algorithms for the structural and parametric identification of linear discrete multivariable systems |1||2|, requires an amount of storage roughly proportional to the square of the system order so that the actual capabilities of many computers can be exceeded when LSS whose order ranges in the hundreds, as it is common in ecosystems, are considered. Similar considerations can be repeated for what concerns the required computation time.

It can however be noted that often LSS are formed by weakly coupled subsystems and that in these cases it is possible to observe, from a single component (or limited group of components) of the output vector, only the projection of the state on a subspace whose dimension is much lower than the whole system order. Moreover, as it has been shown in |3|, it is possible to completely describe the dynamical behavior between the input of the system and the selected output component(s) by means of a model whose minimal order equals the dimension of the subspace observable from this component(s). It is therefore possible to describe the whole system by means of a set of independent models uniquely determined once a partition of the output components has been selected. As it is obvious, the sum of the dimensions of these models, because of their independence, will be generally greater than the order of the whole system also if the dimension of every partial model is minimal.

In this paper, a procedure for the structural and parametric identification of these partial models in linear discrete multivariable systems is described, first in the deterministic case, then in the case of uncorrelated zero-mean additive noise on the data with known statistics (these can also be unknown when the same amount of noise is present on the input/output sequences). The main features of the proposed procedure are the following:

1) The input of the system can be a normal operating sequence (subject only to well-known conditions of complete 'mode excitation'); the use of such unpractical inputs as white noise or impulsive functions is avoided.

2) The initial state of the system needs not to be zero or known and can be easily estimated after the identification of the model.

3) The structure of the identified model is canonical both for models describing the link between a single output component or more components (multivariable case) and the input.

4) The algorithm works directly on the input/output sequences of the system; no intermediate steps (such as, for instance, the computation of the Markov parameters) are required.

These characteristics mean, in practice, a good deal of simplicity in the implementation of the procedure. The method can also be applied, in a deterministic context, to obtain partial models starting from a complete description of a LSS.

II. STATEMENT OF THE PROBLEM

Let consider a linear, completely reachable (1), discrete system described by the usual equations

$$x(k+1) = \mathbf{F}x(k) + \mathbf{G}u(k) \qquad (2.1a)$$

$$y(k) = \mathbf{H}x(k) \qquad (2.1b)$$

where $x(k)=(x_{k,1}\, x_{k,2} \ldots x_{k,n})^T$ belongs to the n-dimensional state space $\mathcal{X}$, $u(k)=(u_{k,1}\, u_{k,2} \ldots u_{k,r})^T$ to the r-dimensional input space $\mathcal{U}$, $y(k)=(y_{k,1}\, y_{k,2} \ldots y_{k,m})^T$ to the m-dimensional output space $\mathcal{Y}$ and $\mathbf{F}, \mathbf{G}, \mathbf{H}$ are constant real-valued matrices of proper size. In the following the index k will be omitted (i.e., for instance, $y=(y_1 \ldots y_m)^T$) when not relevant.

The identification procedure proposed in this paper can be reduced to the solution of the following problems:

Problem 1 : For every subset $y_p, y_q, \ldots, y_s$ $(1 \leqslant p,q,\ldots,s \leqslant m)$ of output components, prove the existence of a partition of system (2.1) into two subsystems S_1 and S_2 (Fig. 1) such that S_1 is a minimal order subsystem completely describing the dynamical behavior between the input and $y_p, \ldots, y_s$ and is not influenced by S_2.

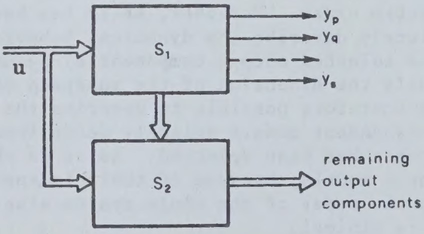


Fig. 1 - System Partition

---

(1) If det $\mathbf{F} \neq 0$, complete controllability can be assumed.

Problem 2 : Find an equivalent, completely identifiable, input/output representation of the type

$$L(z)\ y_1(k) = M(z)\ u(k) \tag{2.2}$$

where z is the unitary delay operator and $y_1=(y_p , y_q , \dots , y_s)^T$, for S_1 .

Problem 3 : Develop an algorithm for the direct structural and parametric identification of representation (2.2) from input/output sequences of system (2.1) corrupted by additive, zero mean, uncorrelated noise.

In fact, the solution of problem 1 establishes the existence of a minimal order representation describing only the subsystem (S_1) to be identified while the solution of problem 2, allowing an easy passage from a state-space representation of S_1 to an input/output description of type (2.2) and *viceversa*, greatly simplifies the identification problem that is reduced, in this way, to the solution of problem 3. The real key of the procedure, however, lies in the quasi-canonical (block-companion) form used to describe S_1 in the solution of problem 1 which yields to a natural, straightforward solution of problems 2 and 3.

The material is arranged as follows. In section 3 the solution of problems 1 and 2 is given while in section 4 problem 3 is considered; in these sections a subset of output components with a single element is considered. The extension to the general case is given in the sections 5 and 6 and in section 7 some concluding remarks are reported.

III. PARTITION OF THE SYSTEM

A subset of output components given by a single element, y_j , will be considered here and in the next section.

Definition 1 : Let **v** be a vector belonging to the n-dimensional space $\mathscr{X}$ and $\mathbf{L}:\mathscr{X}\rightarrow\mathscr{X}$ a linear operator on $\mathscr{X}$. The cyclic invariant subspace $\mathscr{S}\subseteq\mathscr{X}$ spanned by the vectors

$$\mathbf{v},\ \mathbf{L}\mathbf{v},\ \mathbf{L}^2\mathbf{v},\ \dots\dots$$

will be said to be *generated* by **v** (with respect to **L**).

Let

$$\mathbf{H}=\begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \vdots \\ \mathbf{h}_m^T \end{bmatrix}. \tag{3.1}$$

Then the solution of problem 1 is a direct consequence of the proof of the following theorem.

Theorem 1 |3|*:* The minimal order of a model completely describing the dynamical behavior between the input of system (2.1) and its j-th output component, y_j , equals the dimension of the cyclic subspace generated by $\mathbf{h}_j$ with respect to the linear operator $\mathbf{F}^T$.

Proof: A constructive proof will be given. To this purpose construct the vector sequence

$$\mathbf{h}_j\quad \mathbf{F}^T\mathbf{h}_j\ \dots\dots\dots\ \mathbf{F}^{T^{\nu_j-1}}\mathbf{h}_j\quad \mathbf{F}^{T^{\nu_j}}\mathbf{h}_j\dots \tag{3.2}$$

until a dependent vector is found; let it be $\mathbf{F}^{T^{\nu_j}}\mathbf{h}_j$. Then it is possible to express this vector as a linear combination of previous ones, i.e.

$$\mathbf{F}^{T\nu_j}\mathbf{h}_j = \alpha_{j,1}\mathbf{h}_j + \alpha_{j,2}\mathbf{F}^T\mathbf{h}_j + \ldots\ldots\ldots\ldots + \alpha_{j,\nu_j}\mathbf{F}^{T\nu_j - 1}\mathbf{h}_j \quad . \tag{3.3}$$

Consider now the n × n matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{h}_j^T \\ \vdots \\ \mathbf{h}_j^T\mathbf{F}^{\nu_j - 1} \\ \hline \mathbf{Q}_P \end{bmatrix} \tag{3.4}$$

where $\mathbf{Q}_P$ can be any matrix such that $\mathbf{P}$ is nonsingular. Let indicate with $\mathbf{p}_j$ the ν_j-th column of $\mathbf{P}^{-1}$. Then construct the matrix

$$\mathbf{T} = \left\{ \mathbf{p}_j \quad \mathbf{F}\mathbf{p}_j \ \ldots\ldots\ \mathbf{F}^{\nu_j - 1}\mathbf{p}_j \,\middle|\, \mathbf{Q}_T \right\} \tag{3.5}$$

where $\mathbf{Q}_T$ can be any matrix such that $\mathbf{T}$ is nonsingular. By performing on system (2.1) the state transformation $\mathbf{w} = \mathbf{T}^{-1}\mathbf{x}$, the new equivalent equations are obtained:

$$\mathbf{w}(k+1) = \mathbf{A}\mathbf{w}(k) + \mathbf{B}\mathbf{u}(k) \tag{3.6}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{w}(k) \tag{3.7}$$

where the triple $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ exibits the structure:

$$\mathbf{A} = \mathbf{T}^{-1}\mathbf{F}\mathbf{T} = \left[\begin{array}{c|c} \mathbf{A}_j & \\ \hline \mathbf{R} & \mathbf{S} \end{array}\right] \tag{3.8}$$

$$\mathbf{A}_j = \begin{bmatrix} 0 \ \cdots\cdots\cdots \ 0 & \alpha_{j,1} \\ & \alpha_{j,2} \\ \mathbf{I}_{\nu_j - 1} & \vdots \\ & \vdots \\ & \alpha_{j,\nu_j} \end{bmatrix} \tag{3.9}$$

$$\mathbf{C} = \mathbf{H}\mathbf{T} = \begin{bmatrix} \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_{\nu_j - 1}^T \\ \nu_j\text{-th identity row} \\ \mathbf{c}_{\nu_j + 1}^T \\ \vdots \\ \mathbf{c}_m^T \end{bmatrix} \tag{3.10}$$

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{G} = \begin{bmatrix} \mathbf{B}_j \\ \mathbf{B}_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_{\nu_j}^T \\ \hline \mathbf{b}_{\nu_j + 1}^T \\ \vdots \\ \mathbf{b}_n^T \end{bmatrix} = \begin{bmatrix} b_{i\ell} \end{bmatrix} \tag{3.11}$$

These results follow directly, in an obvious duality context, from |4|, |5|. Here the matrices $\mathbf{R}$ and $\mathbf{S}$ do not exibit any particular structure in that $\mathbf{Q}_T$ in (3.5) is quite general; these matrices do not play, however, any further role. The state space of the original system has been in this way decomposed into the direct sum of two subspaces $\mathscr{S}_1$ and $\mathscr{S}_2$ and $\mathscr{S}_1$ is invariant with respect to the linear operator $\mathbf{A}^T$; the system is therefore decomposed into two subsystems S_1 and S_2 according to Fig. 1. Writing $\mathbf{w} = (\mathbf{w}_j^T, \mathbf{w}_r^T)$, where $\mathbf{w}_j = (w_1\ w_2 \ldots w_{\nu_j})^T$, the obtained equations for S_1 are

$$\mathbf{w}_j(k+1) = \mathbf{A}_j\mathbf{w}_j(k) + \mathbf{B}_j\mathbf{u}(k) \tag{3.12a}$$

$$\mathbf{y}_j = w_{\nu_j} \quad . \tag{3.12b}$$

Since the reachable subspace of a linear system is invariant with respect to a change of coordinates, $\mathscr{S}_1$ is completely reachable. However $\mathscr{S}_1$ is also invariant with respect to $\mathbf{A}^T$ so that the couple $(\mathbf{A}_j, \mathbf{B}_j)$, i.e. subsystem S_1, is completely reachable and, because of the canonical structure of equations (3.12), completely observable. The order of S_1 is therefore minimal. ◁

Owing to the canonical representation (3.12) of S_1, an input/output description follows immediately. Precisely, an equivalent representation of type (2.2) is given by

$$(z^{\nu_j} - \alpha_{j,\nu_j} z^{\nu_j - 1} - \ldots - \alpha_{j,1}) y_{k,j} = \sum_{1}^{r} {}_i (z^{\nu_j - 1} b_{\nu_j,i} + \ldots + b_{1,i}) u_{k,i} \tag{3.13}$$

A proof can be found in |6|, p. 177; relation (3.13) can however be immediately deduced from the graph of S_1 (see Fig. 2).

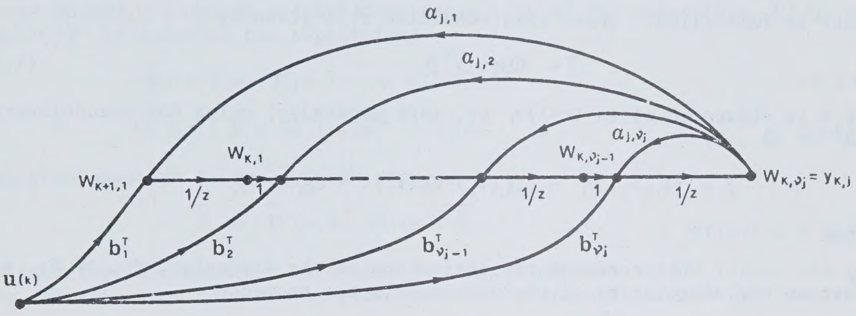


Fig. 2 - Graph of representation (3.12)

As it follows from (3.9), (3.10), (3.11), (3.12) and (3.13), all the nontrivial coefficients appearing in the canonical state-space representation (3.12) are present also in the input/output description (3.13) so that (3.12) can be immediately deduced from (3.13). This property is practically relevant in that, since the difference representation (3.13) directly links the input to one output component, its coefficients can be obtained by means of simple algorithms from input/output sequences.

IV. IDENTIFICATION ALGORITHM

Let introduce the following notation:

$$\mathbf{H}_{p,q}(v_{k,j}) = \begin{bmatrix} v_{k,j} & v_{k+1,j} & \cdots & v_{k+q-1,j} \\ v_{k+1,j} & v_{k+2,j} & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ v_{k+p-1,j} & \cdots & \cdots & v_{k+p+q-2,j} \end{bmatrix} . \tag{4.1}$$

$\mathbf{H}_{p,q}(v_{k,j})$ denotes therefore the p × q Hankel matrix whose first element is given by the j-th component of the vector v(k). Consider now the composite matrix:

$$\mathbf{Q}_j(\nu_j) = \{ \mathbf{H}_y \mid \mathbf{H}_u \} = \{ \mathbf{H}_{N,\nu_j}(y_{k,j}) \mid \mathbf{H}_{N,\nu_j}(u_{k,1}) \ldots \mathbf{H}_{N,\nu_j}(u_{k,r}) \} \tag{4.2}$$

where N can be every integer satisfying the condition $N > (r+1)\nu_j$. Then the input/output relation (3.13) can be written N times in the form

$$\tilde{\mathbf{y}}_j = \mathbf{Q}_j(\nu_j) \gamma_j \tag{4.3}$$

where

$$\tilde{\mathbf{y}}_j = (y_{k+\nu_j,j} \quad y_{k+\nu_j+1,j} \quad \ldots\ldots\ldots \quad y_{k+\nu_j+N-1,j})^T \tag{4.4}$$

$$\gamma_j = (\alpha_{j,1}\ \alpha_{j,2}\ \dots \alpha_{j,\nu_j} \mid b_{1,1} \dots b_{\nu_j,1} \mid \dots \mid b_{1,r}\ \dots\ b_{\nu_j,r})^T . \tag{4.5}$$

As it is well-known |7||8|, the complete identification of a system can be obtained only if the input sequence is 'completely exciting'. Here we only note that, in order to obtain the parameters vector γ_j from (4.3), the condition

$$\operatorname{rank}\ (\mathbf{Q}_j\ (\nu_j)) = (r+1)\nu_j \tag{4.6}$$

must be fullfilled. Under this condition γ_j is given by

$$\gamma_j = \mathbf{Q}_j(\nu_j)^{-1}\,\tilde{y}_j \tag{4.7}$$

if N is chosen equal to $(r+1)\nu_j$ or, more generally, using the pseudoinverse $\mathbf{Q}_j^{\dagger}$ of $\mathbf{Q}_j$, by

$$\gamma_j = \mathbf{Q}_j^{\dagger}(\nu_j)\,\tilde{y}_j = \{\mathbf{Q}_j^T(\nu_j)\ \mathbf{Q}_j(\nu_j)\}^{-1}\ \mathbf{Q}_j^T(\nu_j)\,\tilde{y}_j \tag{4.8}$$

when $N > (r+1)\nu_j$.

For what concerns the estimation of the dimension, ν_j, of S_1, a test on the singularity of the matrices in the sequence

$$\mathbf{P}_j(2) \quad \mathbf{P}_j(3) \quad \dots\dots \quad \mathbf{P}_j(\nu_j) \quad \dots . \tag{4.9}$$

where

$$\mathbf{P}_j(h) = \mathbf{Q}_j^T(h)\ \ \mathbf{Q}_j(h) \tag{4.10}$$

can be performed. Obviously, also for the order determination, the input must satisfy condition (4.6). If $\mathbf{P}_j(\mu_j)$ is the first singular matrix in (4.9), then $\nu_j = \mu_j - 1$.

Let consider now an additive uncorrelated noise with zero mean on the input/output sequences of the system, i.e. let

$$\begin{aligned} y^*_{k,j} &= y_{k,j} + d(y_{k,j}) \\ u^*_{k,1} &= u_{k,1} + d(u_{k,1}) \\ &\vdots \\ u^*_{k,r} &= u_{k,r} + d(u_{k,r}) \quad . \end{aligned} \tag{4.11}$$

(In (4.11) and in the following starred quantities are noisy ones) In this case the least-squares formula (4.8) gives, as it is well-known |9||10|, a biased estimate of γ_j also for $N \to \infty$. The origin of the bias in (4.8) can however be easily detected and eliminated if the statistics of the noise are known. In fact, from the expression of $\mathbf{P}_j^*(\nu_j)$, since the noise has zero mean and is uncorrelated with the input/output sequences of the system, it follows that |1|

$$\operatorname*{plim}_{N\to\infty}\ \mathbf{P}_j^*(\nu_j) = \lim_{N\to\infty} \mathbf{P}_j(\nu_j) + N\,\mathbf{R}\{\mathbf{d}_j(k)\} \tag{4.12}$$

where $\mathbf{R}\{\mathbf{d}_j(k)\}$ is the covariance matrix of the noise vector

$$\begin{aligned} \mathbf{d}_j(k) = \{&d(y_{k,j})\ d(y_{k+1,j})\ \dots\ d(y_{k+\nu_j-1,j}) \mid d(u_{k,1})\ \dots\ d(u_{k+\nu_j-1,1}) \mid \dots \\ &\dots \mid d(u_{k,r})\ \dots\ d(u_{k+\nu_j-1,r})\} \end{aligned} \tag{4.13}$$

and

$$\operatorname*{plim}_{N\to\infty} \mathbf{Q}_j^{T*}(\nu_j)\, \tilde{\mathbf{y}}_j^{*} = \lim_{N\to\infty} \mathbf{Q}_j^{T}(\nu_j)\, \tilde{\mathbf{y}}_j + \mathbf{r}_j(k) \qquad (4.14)$$

where

$$\mathbf{r}_j(k) = \lim_{N\to\infty} \{ \textstyle\sum_k^N (d(y_{i,j})d(y_{i\text{-}\nu_j,j})) \ldots\ldots \sum_k^N (d(y_{i\text{-}\nu_{j-1},j})d(y_{i\text{-}\nu_j,j})) \mid \ldots\ldots$$
$$\ldots \mid \ldots\ldots\ldots \textstyle\sum_k^N (d(u_{i\text{-}\nu_{j-1},r})d(y_{i\text{-}\nu_j,j})) \}^T .$$

If the statistics of the noise are known it is therefore possible, for N large enough, to obtain a consistent estimation of the quantities $\mathbf{P}_j(\nu_j)$ and $\mathbf{Q}_j^T(\nu_j)\tilde{\mathbf{y}}_j$ by means of the expressions

$$\hat{\mathbf{P}}_j(\nu_j) = \mathbf{P}_j^{*}(\nu_j) - N\, \mathbf{R}\{\, \mathbf{d}_j(k)\} \qquad (4.15)$$

$$\hat{\mathbf{Q}}_j^T(\nu_j)\, \hat{\tilde{\mathbf{y}}}_j = \mathbf{Q}_j^{T*}(\nu_j)\tilde{\mathbf{y}}_j^{*} - \mathbf{r}_j(k) \quad . \qquad (4.16)$$

The "compensated" least-squares estimator

$$\hat{\gamma}_j = \hat{\mathbf{P}}_j(\nu_j)^{-1}\, \hat{\mathbf{Q}}_j(\nu_j)\, \hat{\tilde{\mathbf{y}}}_j \qquad (4.17)$$

is therefore a consistent estimator for the parameters of S_1 . Also the order determination test can be performed on the sequence of the matrices $\mathbf{P}_j$.

Remark 1 : It can be noted that, when the same amount of uncorrelated zero-mean noise is added to the input/output sequences and nonoverlapping sets of data are used, then $\mathbf{R}\{ \mathbf{d}_j(k)\} = \sigma^2 \mathbf{I}$ where σ^2 is the variance of the noise so that

$$\operatorname*{plim}_{N\to\infty} \mathbf{P}_j^{*}(\nu_j) = \lim_{N\to\infty} \mathbf{P}_j(\nu_j) - N\sigma^2 \mathbf{I} \qquad (4.18)$$

$$\operatorname*{plim}_{N\to\infty} \mathbf{Q}_j^{T*}(\nu_j)\, \tilde{\mathbf{y}}_j^{*} = \lim_{N\to\infty} \mathbf{Q}_j^{T}(\nu_j)\, \tilde{\mathbf{y}}_j \qquad (4.19)$$

and (4.17) becomes

$$\hat{\gamma}_j = (\mathbf{P}_j^{*}(\nu_j) - N\sigma^2 \mathbf{I})^{-1}\, \mathbf{Q}_j^{T*}(\nu_j)\tilde{\mathbf{y}}_j^{*}. \qquad (4.20)$$

It is also important to note that, in this case |1||11|, an estimate of $N\sigma^2$ is given by the least eigenvalue of the symmetrical matrix $\mathbf{P}_j^{*}(\mu)$ for $\mu > \nu_j$ so that a consistent estimate of the order and of the parameters of S_1 can be obtained starting from a given upper bound to the order of S_1.

Remark 2 : If the system to be identified is not completely reachable and if its initial state does not belong to the reachability subspace, then the identified partial model will be a nonminimal one in that completely observable but not completely reachable and must therefore be successively reduced by means of standard techniques |12|. This inconvenience, common to all partial or complete identification procedures, is not relevant when the number of available data is large.

Remark 3 : Once the identification of the partial model has been carried out its initial state can be easily estimated. In fact, considering for simplicity noise-free data, the initial state of S_1,w_j, is given by relation (4.21). This relation can easily be obtained from the graph of S_1 (Fig. 2) and extended to the case of noisy data.

$$\begin{bmatrix} w_{K,1} \\ w_{K,2} \\ \vdots \\ w_{K,\nu_j-1} \\ w_{K,\nu_j} \end{bmatrix} = \begin{bmatrix} -\alpha_{j,2} & -\alpha_{j,3} & \cdots & -\alpha_{j,\nu_j} & 1 \\ -\alpha_{j,3} & -\alpha_{j,4} & & 1 & \\ \vdots & & \iddots & & \\ -\alpha_{j,\nu_j} & 1 & & & \\ 1 & & & & \end{bmatrix} \begin{bmatrix} y_{K,j} \\ y_{K+1,j} \\ \vdots \\ y_{K+\nu_j-2,j} \\ y_{K+\nu_j-1,j} \end{bmatrix} - \tag{4.21}$$

$$- \begin{bmatrix} b_2^T & b_3^T & \cdots & b_{\nu_j}^T \\ b_3^T & b_4^T & \iddots & \\ \vdots & \iddots & & \\ b_{\nu_j}^T & & & \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+\nu_j-2) \end{bmatrix}$$

Remark 4 : In practice it is relevant to observe that the identification algorithm can also be applied to a model whose order has been *a priori* fixed. In this case estimators (4.8) and (4.17), (4.20) give a model whose output sequences best fit, for the given order, the real system output sequence according to the least-squares criterion.

SECTIONS V AND VI

These sections are omitted, for lack of space, in the conference version of this paper.

VII. CONCLUSIONS

A partial identification technique for large-scale systems has been considered. The proposed method can prove useful in the identification of systems constituted by many weakly coupled subsystems and relies on the equivalence between block-companion state space representations and difference input/output descriptions. A nice feature of the given algorithm is that it works directly on input/output data and leads to a state space description in a very limited number of steps.

REFERENCES

|1| C. Bonivento and R. Guidorzi, "Parametric Identification of Linear Multivariable Systems", *Preprints of the 1971 JACC*, pp. 342-349, St. Louis.

|2| C. Bonivento, R. Guidorzi and G. Marro, "Irreducible Canonical Realizations from External Data Sequences", *Int. J. of Control*, (to appear).

|3| R. Guidorzi and F. Terragni, "Zeros Determination in Large-Scale Multivariable Systems", *Int. J. of Control*, (to appear).

|4| D.G. Luenberger, "Canonical Forms for Linear Multivariable Systems", *IEEE Trans. on Automatic Control*, vol. AC-12, pp. 290-293, June 1967.

|5| W.M. Wonham and C.D. Johnson, "A Note on the Transformation to Canonical (phase-variable) Form", *IEEE Trans. on Automatic Control*, vol. AC-9, pp. 312-313, July 1964.

|6| R.E. Kalman, "Mathematical Description of Linear Dynamical Systems", *J. SIAM Control*, vol. 1, no. 2, pp.152-192.

|7| M. Aoki and R.M. Staley, "On Input Signal Synthesis in Parametric Identification", *Proc. IV IFAC Congress*, Warsaw, 1969.

|8| K.J. Aström and P. Eykhoff, "System Identification", *Survey Paper, 2nd Prague IFAC Symp. on Identification and Process Parameter Estimation*, Prague, 1970.

|9| R.G. Baender and F.W. Smith, "A Note on the Statistical Estimation of a Hyperplane", *IEEE Trans. on Automatic Control*, vol. AC-13, p. 591, Oct. 1968.

|10| T. Koopmans, "Identification Problems in Economic Model Construction", *Studies in Econometric Methods*, J. Wiley and Sons, 1953.

|12| D.Q. Mayne, "Computational Procedure for the Minimal Realization of Transfer Function Matrices", *Proc. IEE*, no. 115, pp. 1363-1368.

|13| W.A. Wolovich and P.L. Falb, "On the Structure of Multivariable Systems", *SIAM J. Control*, vol. 7, no. 3, pp. 437-451, August 1969.

A FUNCTIONAL APPROACH TO LARGE-SCALE DYNAMICAL SYSTEMS

M. N. RANSOM and R. SAEKS
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

The connections in a large-scale dynamical system are interpreted as a system valued function of a system valued variable, the derivative and various inverses of this function being employed in the solution of system theoretic problems. Moreover, matrix algebraic representations for the "connection functions" are obtained which allow straightforward digital computer implementation of the resultant theory.

INTRODUCTION

An interconnection of components, each of which is a (possibly multiple-input, multiple-output) dynamical system is termed a large-scale dynamical system (LSDS). Traditionally large-scale dynamical systems have been studied from the point of view of the input-output-state relations they present. Consequently the connections of an LSDS have been used as a means of obtaining these relations. An alternate interpretation is to view the connections of an LSDS as defining a mapping of the system components into the external system behavior. That is, let z_i be a matrix of operators modeling the i^{th} subsystem of an LSDS and u and y be vectors of inputs and outputs of the overall system. We then write

$$y = [f(z_1,z_2,\ldots)]u \tag{1}$$

where f is a function determined entirely by the connections between the subsystems. Although f is nonlinear even for linear systems, it is of a very special nature readily studied by matrix algebraic techniques. As such it is feasible to take a functional point of view of the connections of an LSDS. In particular, the standard tools and techniques of differential calculus may be applied to the study of the connections of a large-scale dynamical system, and we may talk about the derivative or the inverse of the connections of such a system. In the present paper such a point of view is adopted with the problem of system sensitivity being formulated in terms of the derivative of f, and the problem of fault isolation and system synthesis being formulated in terms of the inverse and/or generalized inverse of f.

COMPONENT CONNECTION MODEL

In a number of recent papers the authors have adopted the component connection model for LSDS.[1,2,3] Let a_i be the vector of input time functions (or sequences) applied to the i^{th} component (subsystem) and b_i be the vector of output functions measured at the i^{th} component. Symbolically, an LSDS then has the form shown in Figure 1. Here the u and y variables represent the system inputs and outputs, respectively, and $a = \mathrm{col}(a_i)$ and $b = \mathrm{col}(b_i)$. Here the components box represents all of the system's components and the shaded area represents the connectivity structure which consists of adders and scalers. If one now redraws Figure 1 as shown in Figure 2 wherein the components and connectivity structure are separated, it becomes natural to view the connections as a separate multiple input-multiple output component with u and b taken as the inputs and y and a as the outputs. Since the connection elements are all linear and algebraic, the connection model can be represented by a set of algebraic equations of the form

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & 0 \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} \tag{2}$$

Here, for simplicity, we assume that there is no direct connection between the inputs, u, and the outputs, y. A computer algorithm for generating the "L" matrix is given by Prasad and Trauboth.[4,5] The inputs and outputs of the i^{th} component are related by the equation

$$b_i = z_i a_i \tag{3}$$

Here z_i is an operator (differential, difference, etc.) modeling the i^{th} component. More succinctly, we may write

$$b = Za \tag{4}$$

where $Z = \mathrm{diag}(z_i)$. Combining the above equations for the overall dynamical system, T, relating y and u, we obtain

$$y = Tu \tag{5a}$$

where

$$T = L_{21}(1 - ZL_{11})^{-1} ZL_{12} = f(Z) \tag{5b}$$

Example

Consider the active network shown in Figure 3, which is an RC-coupled amplifier driving an inductive load. The connection equations may be written as

$$\begin{bmatrix} v_i \\ v_L \\ i_c \\ v_R \\ E_o \end{bmatrix} = \left[\begin{array}{cccc|c} 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} v_o \\ i_L \\ v_c \\ i_R \\ E_i \end{bmatrix}$$

The component equations are then

$$\begin{bmatrix} v_o \\ i_L \\ v_c \\ i_R \end{bmatrix} = \begin{bmatrix} \mu g(\omega) & 0 & 0 & 0 \\ 0 & 1/j\omega L & 0 & 0 \\ 0 & 0 & 1/j\omega C & 0 \\ 0 & 0 & 0 & 1/R \end{bmatrix} \begin{bmatrix} v_i \\ v_L \\ i_c \\ v_R \end{bmatrix}$$

Sensitivity

The sensitivity of an LSDS to changes in its subsystems can be readily determined using the component connection model. The sensitivity of a function g to a variable x is defined as

$$S_x^g = \frac{\partial g}{\partial x}\frac{x}{g} \tag{6}$$

Consider the calculation of the partial derivative of the system matrix T defined in equation 5a, with respect to a parameter vector α. Using the chain rule,[11] we may write

$$\frac{\partial T}{\partial \alpha} = \frac{\partial f(Z)}{\partial \alpha} = \frac{\partial f(Z)}{\partial Z}\frac{\partial Z}{\partial \alpha} \tag{7}$$

Thus the sensitivity of the overall system can be expressed in terms of the sensitivities of the subsystems. Repeating equation 5b,

$$T = L_{21}(1 - ZL_{11})^{-1}ZL_{12}$$

and invoking the formula

$$\frac{\partial}{\partial y}(1+XA)^{-1} = -(1+XA)^{-1}(\frac{\partial}{\partial y}X)A(1+XA)^{-1} \tag{8}$$

where X is a matrix function of the variable y, and A is a constant matrix, we may write

$$\dot{T} = L_{21}(1-ZL_{11})^{-1}\dot{Z}[L_{11}(1-ZL_{11})^{-1}Z+1]L_{12} \tag{9}$$

where $\dot{T}$ and $\dot{Z}$ represent differentiation with respect to α. Applying the rule[6]

$$\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B) \tag{10}$$

where

$$\mathrm{vec}(A) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \tag{11}$$

A_i being the i^th^ column of A and where $\otimes$ represents the Kronecker product defined by

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & & \\ a_{n1}B & \dots & a_{nm}B \end{bmatrix} \tag{12}$$

we may write

$$\begin{aligned}\dot{\bar{T}} &= \{[L_{11}(1-ZL_{11})^{-1}Z+1]L_{12}\}^T \otimes \{L_{21}(1-ZL_{11})^{-1}\}\dot{\bar{Z}} \\ &= J\,\dot{\bar{Z}}\end{aligned} \tag{13}$$

where $\bar{T} = \mathrm{vec}(T)$ and $\bar{Z} = \mathrm{vec}(Z)$. Equation 12 may be written in the partitioned form

$$\left[\frac{\partial}{\partial a_1}\bar{T} \;\Big|\; \frac{\partial}{\partial a_2}\bar{T} \;\Big|\; \dots \;\Big|\; \frac{\partial}{\partial a_k}\bar{T}\right] = J\left[\frac{\partial}{\partial a_1}\bar{Z} \;\Big|\; \frac{\partial}{\partial a_2}\bar{Z} \;\Big|\; \dots \;\Big|\; \frac{\partial}{\partial a_k}\bar{Z}\right] \tag{14}$$

The sensitivity matrix is then found to be

$$S_a^{\bar{T}} = [S_{a_1}^{\bar{T}} \mid S_{a_2}^{\bar{T}} \mid \dots \mid S_{a_k}^{\bar{T}}] = \hat{T}^{-1}J\dot{\bar{Z}}\hat{a} \tag{15}$$

where $\hat{T} = \mathrm{diag}(\bar{T})$ and $\hat{a} = \mathrm{diag}(a)$.

Fault Isolation

The problem of fault isolation may be states as "Given a set of external measurements, determine the values of the internal components which produced these measurements." In terms of equation 5b this amounts to finding the left inverse of the function f. This problem was treated in reference 3, and its results are summarized here.

Consider the case where all the external measurements (T) are taken at a single frequency. A necessary and sufficient condition for the internal parameters (Z) to be determined exactly from these measurements is for L_{21}^{-L} and L_{12}^{-R} to exist, in which case

$$Z = L_{21}^{-L}\,T\,L_{12}^{-R}\,(1 + L_{11}\,L_{21}^{-L}\,T\,L_{12}^{-R})^{-1} \tag{16}$$

For the case where measurements are made at different frequencies, we assume that the frequency variation of Z is separable from the component values to the extent that Z may be written in the form

$$Z = C(\omega)\;D\;E(\omega) \tag{17}$$

where $C(\omega)$ and $E(\omega)$ are known, non-singular matrix-functions of frequency, and D is the parameter matrix to be determined. Define $L_{21}^{i} = L_{21}C(\omega_i)$ and $L_{12}^{i} = E(\omega_i)L_{12}$. Find matrices K_{2i} and K_{3i}, i=1,2,...,n such that

$$I = \sum_{i=1}^{n} (K_{2i}^{T} \otimes K_{3i})(L_{12}^{iT} \otimes L_{21}^{i}) \tag{18}$$

and define

$$K_2 = \left[\begin{array}{c} K_{21} \\ \hdashline K_{22} \\ \hdashline \vdots \\ \hdashline K_{2n} \end{array}\right] \tag{19}$$

and

$$K_3 = [K_{31} \,\vdots\, K_{32} \,\vdots \ldots \vdots\, K_{3n}] \qquad (20)$$

Further, find a matrix $\bar{K}_1$ such that

$$L_{11} + L_{12}\,\bar{K}_1\,L_{21} = 0 \qquad (21)$$

and define $K_1 = \text{diag}[\bar{K}_1, \bar{K}_1, \ldots, \bar{K}_1]$. Let $T = \text{diag}[T_1, T_2, \ldots T_k]$ where T_i is the measurements made at ω_i. The parameter matrix is then given by

$$D = C^{-1}(\omega_i)K_3(1-TK_1)^{-1}TK_2D^{-1}(\omega_i) \qquad (22)$$

where any ω_i can be used.

Approximate Fault Isolation

For the case where an insufficient number of measurements are made, a left inverse of f in equation 5b does not exist. In this case a generalized inverse[7,8] of f will reveal a solution, $\bar{Z}$, which approximates the best least squares estimate of Z. The solution to this problem when all measurements are made at a single frequency is treated in reference 9. Its solution is an estimation to the problem, "Given a matrix of measurments, T, and the connection matrices, L_{ij}, find a matrix, Z, such that T = f(Z) and for all $\bar{Z}$ satisfying $T = f(\bar{Z})$, $||Z-Z_o|| \leqq ||\bar{Z}-Z_o||$," where Z_o is a specified nominal component matrix. If one assumes that small errors from nominal are more likely than large errors, as in the case of components drifting out of tolerance, the norm condition assures that we have the "best" estimate to the actual component values.

The solution to this problem is facilitated by the fact that f can be decomposed into a linear function and a nonlinear but invertible function. Namely,

$$f = hg \qquad (23a)$$

where

$$T = h(R) = L_{21}\,R\,L_{12} \qquad (23b)$$

and

$$R = g(Z) = (1 - ZL_{11})^{-1}Z \qquad (23c)$$

Computer algorithms for determining the generalized inverse of linear functions are currently available.[10] In order for the generalized inverse of h to represent a generalized inverse of f, h is first scaled as follows. Let $\dot{R}$ be a matrix whose i,j element is the partial derivative of the i^{th} component of vec(R) with respect fo the j^{th} component of vec(Z). Using the procedure used to obtain the sensitivity matrix, we obtain

$$\dot{R} = [L_{11}(1-ZL_{11})^{-1}Z + 1] \otimes [(1-ZL_{11})^{-1}] \qquad (24)$$

Further, let W be a diagonal matrix whose i,i element is, according to the designer's judgment, inversely related to the amount of change the i^{th} element of vec(Z) is likely to make from its nominal value. We then define

$$\tilde{h} = (L_{12}^{T} \otimes L_{21}) \dot{R} \; W \tag{25}$$

Let T_o be the matrix of nominal external measurements. Our estimate of Z is then

$$\bar{Z} = (1 + \bar{R}L_{11})^{-1}\bar{R} \tag{26}$$

where

$$\bar{R} = R_o + \dot{R} \; W \; \tilde{h}^{-G} \; (T - T_o) \tag{27}$$

$R_o = g(Z_o)$, and where -G indicates generalized inverse.

Synthesis.

A problem closely related to that of fault isolation is the problem of synthesis using a given topology. Here one wishes to find a set of component values, Z, which will give a given set of external measurements, T. This amounts to taking the right inverse of f of equation 5b. For the case where Z is exactly determined by the matrix, T, the procedure in the section on fault isolation may be used. Of more importance, however, is the case where the system is overspecified so that no set of parameters, Z, will satisfy all the requirements. One property of a generalized inverse of f is that if no solution to f(Z) = T can be found, the generalized inverse will present a solution, $\bar{Z}$, which minimizes the quantity $||T-f(Z)||$. Thus the generalized inverse will give the best least squares approximation to the ideal system response. The procedure given in the section on approximate fault isolation, however, can only be used when all of the system specifications are given at a single frequency. Even then, the generalized inverse can give negative component values. An extension of the procedure to the multi-frequent case along with a technique for obtaining positive component values is currently under study.

REFERENCES

1. Singh, S.P., Ph.D. Thesis, University of Notre Dame, 1971.

2. Liu, R.-W., "Convergent Systems," IEEE Trans. on Automatic Control, Vol. AC-13, 1968, pp. 384-391.

3. Saeks, R., Singh, S.P., Liu, R.-W., "Fault Isolation via Components Simulation," IEEE Trans. on Circuit Theory, Nov., 1972, (to appear).

4. Trauboth, H. and Prasad, N., "MARSYAS - A Software System for the Digital Simulation of Physical Systems," Proc. Spring Joint Computer Conf., 1970, pp. 223-235.

5. Prasad, N., and Reiss, J., "The Digital Simulation of Interconnected Systems," Proc. Conf. of the International Assoc. of Cybernetics, Belgium, 1970.

6. Neudecker, J., "Some Theorems on Matrix Differentiation with Special Reference to Knonecker Matrix Products," Amer. Stat. Assoc. Journal, Sept. 1969, pp. 953-963.

7. Penrose, R., "On the Best Approximate Solutions of Linear Matrix Equations," Proc. Cambridge Phil. Soc., Vol. 52, 1956, pp. 17-19.

8. Rao, C.R. and Mitra, S.K., Generalized Inverse of Matrices and Its Applications, New York: John Wiley & Sons, Inc., 1971.

9. Ransom, M.N., and Saeks, R., "Fault Isolation with Insufficient Measurements," Tech. Memo. EE-7209, University of Notre Dame.

10. Boot, J.C.G., "The Computation of the Generalized Inverses of Singular Matrices," Amer. Math. Monthly, vol. 70, 1963, pp. 302-303.

11. Fleming, W.H., Functions of Several Variables, Reading: Addison-Wesley, 1965.

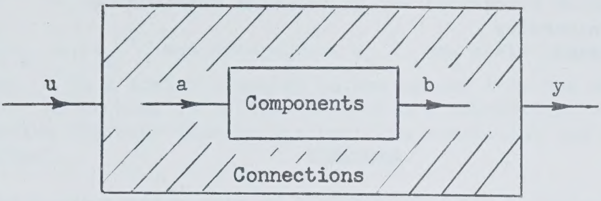

u
a
Components
b
y
Connections

Figure 1.

General form for a large scale system.

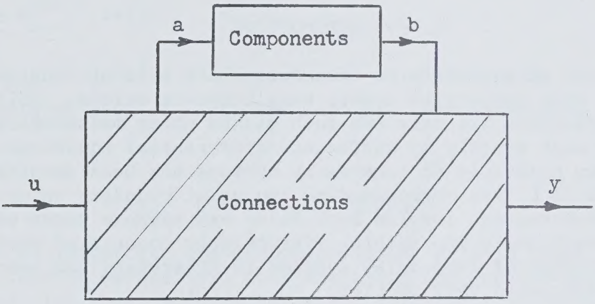

a
Components
b
u
Connections
y

Figure 2.

Figure 1 redrawn in vector notation with components and connections separated.

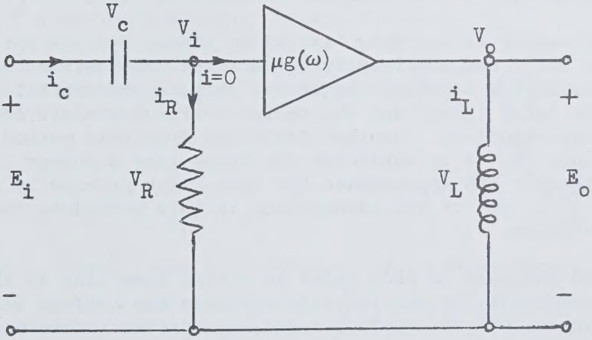

V_c
V_i
i_c
$i=0$
i_R
$\mu g(\omega)$
V_o
i_L
+
+
E_i
V_R
V_L
E_o
−
−

Figure 3.

R-C coupled amplifier with an inductive load.

SUBOPTIMAL CONTROL OF COUPLED TIME DELAY SYSTEMS

MOHAMMAD JAMSHIDI
Department of Electrical and Electronics Engineering
Pahlavi University
Shiraz, Iran

ABSTRACT

The concept of coupling is extended to time-delay systems in this paper. The suboptimal control is a MacLaurin's series expansion of the coupling parameter ε which has both open and closed loop terms. The design method with its computational procedure is given for nonlinear plants. The method is also applicable to linear plants. A numerical example is included.

I. INTRODUCTION

The control of systems with time-delay have been of considerable concern, because many industrial plants have inherent delays. Rolling mills and chemical reactors are only two such plants whose mathematical models are described most exactly by difference-differential equations. The extension of maximum principle to time-delay systems has been developed by Kharatishivili [1]. The corresponding two point boundary value (TPBV) problem for such systems involve both delay and advance terms whose exact solution is practically impossible. The primary concern of most proposed methods of solution of time-delay systems in literature has been to avoid solving the associated TPBV problem.

For linear time-delay processes with quadratic cost functional Krasowskii [2] has developed expressions for the optimal cost function and the controller. The method of Krasowskii has been formulated more conveniently by reducing the control of the time-delay system into the solution of a set of successively coupled partial differential equations [3].

The above method is not only limited to linear systems but as also mentioned in [4], it is computationally cumbersome. Another method is the sensitivity approach [5], in which the state, costate and control are expanded in terms of the delay itself and the series coefficients are obtained by non-delay system solutions. Another deviation from this method Sannuti [6] and Inoue, et.al. [5] is to subdivide the delay into a number of equally spaced sub-intervals and approximate the time-delay problem by a singularly perturbed one [7]. One of the assumptions in this method is that the delay is short in duration.

The method proposed in this paper is on the same line as the sensitivity approach of Inoue, et.al. [5]. It extendes the optimal control method for coupled systems [8] to coupled time-delay systems. Consider a set of m coupled time-delay systems,

$$\dot{z}_1 = h_1(z_1, w_1, \varepsilon c_1(z, z(t-\lambda), w, w(t-\lambda), t), t)$$

$$\dot{z}_2=h_2(z_2,w_2,\varepsilon c_2(z,z(t-\lambda),w,w(t-\lambda),t),t) \tag{1}$$

$$\dot{z}_m=h_m(z_m,w_m,\varepsilon c_m(z,z(t-\lambda),w,w(t-\lambda),t),t)$$

where $z= z_1 z_2 \ldots z_m{}^T$ and $w= w_1 w_2 \ldots w_m{}^T$, are state and control vectors, respectively, ε is a scalar coupling parameter and λ is the delay, not necessary short. The problem is to find a control w composed of w_i, $i=1,2,\ldots,m$ which satisfies (1) with appropriate boundary conditions and minimizes a cost functional,

$$J = \int_{t_o}^{t_f} \sum_{i=1}^{m} L_i(z_i,w_i,t)dt. \tag{2}$$

Assuming sufficient continuity and differentiability of w with respect to ε, the McLaurin's series expansion for w_i has the form,

$$w_i \simeq \sum_{j=1}^{K} \frac{\varepsilon^j}{j!} w_i^{(j)}, \quad i=1,2,\ldots,m \tag{3}$$

where $w_i^{(j)} \underset{=}{\Delta} \lim_{\varepsilon\to 0} \partial^j w_i/\partial\varepsilon^j$. The computation of coefficients of series (3), as in the non-delayed case 8 involves uncoupled and unretarted subsystem solutions.

II. SUB-OPTIMAL CONTROL

In order to illustrate the design procedure of this paper it is assumed, without any loss of generality, that there are two sub-systems involved in (1), i.e. m=2, let $z_1=x$, $z_2=y$, $w_1=u$, $w_2=v$, $c_1(.)=a(.)$, $c_2(.)=b(.)$, $h_1=f$ and $h_2=g$, then the state model equations for the "subsystem x" and "subsystem y" become,

$$\dot{x} = f(x,u,\alpha) = f(x,u,\varepsilon a(x,x', y,y', u,u', v,v')) \tag{4a}$$

$$\dot{y} = g(y,v,\beta) = g(y,v,\varepsilon b(x,x', y,y', u,u', v,v')) \tag{4b}$$

where x and y have, respectively dimensionalities of n_1 and n_2 and the prime on the variables in defind above (see Appendix for symbols). The initial functions for (4) are,

$$x(\tau) = \phi(\tau), \quad t_o-\lambda \leq \tau \leq t_o \tag{5a}$$

$$y(\tau) = \psi(\tau), \quad t_o-\lambda \leq \tau \leq t_o. \tag{5b}$$

Assuming that $L_1(.)=F(.)$ and $L_2(.)=G(.)$ in (2), the necessary conditions for the optimality of the coupled time-delay systems (4)-(5) reduces to the following two time-delay and advance sets of two-point boundary-value problems [1],

$$\dot{x} = H_P^T = f \tag{6a}$$

$$\dot{p} = -H_x^T-H_{x'}^T(s)=F_x^T-(f_x+\varepsilon f_\alpha a_x)^T P-\varepsilon(g_\beta b_x)^T q-\varepsilon\{(f_\alpha a_{x'})^T P+(g_\beta b_{x'})^T q\}\big|_s \cdots \quad t\in I_1 \tag{6b}$$

$$= -H_x^T = F_x^T - (f_x + \varepsilon f_\alpha a_x)^T P - \varepsilon (g_\beta b_x)^T q \;\ldots\quad t \in I_2 \tag{6c}$$

$$0 = H_u^T + H_{u'}^T(s) = -F_u^T + (f_u + \varepsilon f_\alpha a_u)^T P + \varepsilon (g_\beta b_u)^T q + \varepsilon\{(f_\alpha a_{u'})^T P$$

$$+ (g_\beta b_{u'})^T q \}\big|_s ,\ldots\quad t \in I_1 \tag{6d}$$

$$= H_u^T = -F_u^T + (f_u + \varepsilon f_\alpha a_u)^T P + \varepsilon (g_\beta b_u)^T q \;\ldots\quad t \in I_2 \tag{6e}$$

$$\dot{y} = H_q^T = g \tag{7a}$$

$$\dot{q} = -H_y^T - H_y^T(s) = G_y^T - (g_y + \varepsilon g_\beta b_y)^T q - \varepsilon (f_\alpha a_y)^T P$$

$$-\varepsilon\{(g_\beta b_{y'})^T q + (f_\alpha a_{y'})^T P\}\big|_s \ldots\quad t \in I_1 \tag{7b}$$

$$= -H_y^T = G_y^T - (g_y + \varepsilon g_\beta b_y)^T q - \varepsilon (f_\alpha a_y)^T P \;\ldots\quad t \in I_2 \tag{7c}$$

$$0 = H_v^T + H_{v'}^T(s) = -G_v^T + (g_v + \varepsilon g_\beta b_v)^T q + \varepsilon (f_\alpha a_v)^T P$$

$$+\varepsilon\{(g_\beta b_{v'})^T q + (f_\alpha a_{v'})^T P\}\big|_s \ldots\quad t \in I_1 \tag{7d}$$

$$= H_v^T = -G_v^T + (g_v + \varepsilon g_\beta b_v)^T q + \varepsilon (f_\alpha a_v)^T P \;\ldots\quad t \in I_2 \tag{7e}$$

where I_1: $t_o \leq t \leq t_f - \lambda$, $\quad I_2$: $t_f - \lambda \leq t \leq t_f$,

$H = -F - G + P^T f + q^T g$ is the Hamiltonian function, p and q are, respectively, costates of x and y. The final values of costates $P(t_f)$ and $q(t_f)$ are zero vectors. The following assertions, similar to non-delayed case considered by [8], gurantees the evaluation of all coefficients of series (3) by virtue of subsystems computations only.

Assertion 1: At $\varepsilon=0$ the necessary conditions (6) and (7) reduce to two independent TPBV problem for the two un-coupled un-retarted systems.

Proof: At $\varepsilon=0$, none of the functions f, f_x and f_v depend on x', y, y', u', v, and v'. Similarly, none of the functions g, g_y and g_v depend on y', x, x', v', v and u'. Thus,

$$\dot{x} = f, \quad x(t_o) = \phi(t_o) \tag{8a}$$

$$\dot{p} = F_x^T - f_x^T p, \quad p(t_f) = 0 \tag{8b}$$

$$0 = -F_u^T + f_u^T p \tag{8c}$$

$$\dot{y} = g, \quad y(t_o) = \psi(t_o) \tag{9a}$$

$$\dot{q} = G_y^T - g_y^T q, \quad q(t_f) = 0 \tag{9b}$$

$$0 = -G_v^T + g_v^T q. \tag{9c}$$

The solution of TPBV problems (8) and (9) provides $u^{(0)}$ and $v^{(0)}$ which are the zeroth-order terms of series in (3).

Assertion 2: If the solution from the necessary conditions for terms j=0, 1,2,...,i-1 are known, the necessary conditions for the ith terms reduces to two independent un-retarted TPBV problems. Furtheremore, the homogeneous parts of these problems remains unchanged for $i \geq 1$.

Proof of this assertion is similar to Assertion 1 given also in 9

III. COMPUTATIONAL PROCEDURE

For an Nth-order approximation the following steps are required to obtain the sub-optimal control,

Stage 1: Calculate the zeroth-order (non-delayed) responses from (8) and (9)

Stage 2: (a) solve $n_1 x n_1$ and $n_2 x n_2$ matrix Riccati equations,

$$\dot{K}_x = -K_x A_x - A_x^T K_x + K_x S_x K_x - Q_x, \quad K_x(t_f)=0 \tag{10}$$

$$\dot{K}_y = -K_y A_y - A_y^T K_y + K_y S_y K_y - Q_y, \quad K_y(t_f)=0 \tag{11}$$

(b) solve the linear adjoint equations,

$$\dot{g}_x^1 = A_x - S_x K_x)^T g_x^1 - K_x L_x^1 + \ell_x^1 \; \ldots. \qquad t \in I_1 \tag{12a}$$

$$= -(A_x - S_x K_x)^T g_x^1 - K_x L_x^2 + \ell_x^2 \; \ldots \qquad t \in I_2 \tag{12b}$$

$$\dot{g}_y^1 = -(A_y - S_y K_y)^T g_y^1 - K_y L_y^1 + \ell_y^1 \; \ldots \qquad t \in I_1 \tag{13a}$$

$$= -(A_y - S_y K_y)^T g_y^1 - K_y L_y^2 + \ell_y^2 \; \ldots \qquad t \in I_2 \tag{13b}$$

and

$$g_x^1(t_f) = g_y^1(t_f) = 0.$$

(c) obtain $x^{(1)}$ and $y^{(1)}$ from,

$$\dot{x}^{(1)} = (A_x - S_x K_x) x^{(1)} - S_x g_x^1 + L_x^1 \tag{14}$$

$$\dot{y}^{(1)} = (A_y - S_y K_y) y^{(1)} - S_y g_y^1 + L_y^1 \tag{15}$$

with $x^{(1)}(t_o) = y^{(1)}(t_o) = 0$ and $u^{(1)}$ and $v^{(1)}$ are obtained from,

$$u^{(1)} = -(\Gamma_x + \Pi_x K_x) x^{(1)} - \eta_x^1 - \Pi_x g_x^1 \; \ldots\ldots.. \quad t \in I_1 \tag{16a}$$

$$= -(\Gamma_x + \Pi_x K_x) x^{(1)} - \eta_x^2 \; \ldots\ldots.. \quad t \in I_2 \tag{16b}$$

$$v^{(1)} = -(\Gamma_y + \Pi_y K_y) y^{(1)} - \eta_y^1 - \Pi_y g_y^1 \; \ldots\ldots.. \quad t \in I_1 \tag{17a}$$

$$= -(\Gamma_y + \Pi_y K_y) y^{(1)} - \eta_y^2 \; \ldots\ldots.. \quad t \in I_2 \tag{17b}$$

expressions for A_x, S_x, Q_x, Γ_x, Π_x, L_x^1, L_x^2, ℓ_x^1, ℓ_x^2, n_x^1, n_x^2 etc. are given in the appendix.

Stage j (j>2):

Solve the linear adjoint equations similar to (12) and (13) for g_x^j and g_y^j using the same Riccati solution as in (10) and (11),

$$\dot{x}^{(j)}=(A_x-S_xK_x)x^{(j)}-S_xg_x^j+L_x^j \tag{18}$$

$$\dot{y}^{(j)}=(A_y-S_yK_y)y^{(j)}-S_yg_y^j+L_y^j \tag{19}$$

with a similar expressions for $v^{(j)}$ and $u^{(j)}$ as in (16) and (17).

Note that based on the expressions for $u^{(i)}$ and $v^{(i)}$ in (16) and (17) the control structure has the following general form,

$$u=-\alpha_x x+\sum_{i=1}^{N}\frac{\varepsilon^i}{i!}\beta_{xi} \quad \ldots\ldots\ldots \quad t\in I_1 \tag{20a}$$

$$=-\alpha_x x+\sum_{i=1}^{N}\frac{\varepsilon^i}{i!}\gamma_{xi} \quad \ldots\ldots\ldots \quad t\in I_2 \tag{20b}$$

$$v=-\alpha_y y+\sum_{i=1}^{N}\frac{\varepsilon^i}{i!}\beta_{yi} \quad \ldots\ldots\ldots \quad t\in I_1 \tag{21}$$

$$=\alpha_y y+\sum_{i=1}^{N}\frac{\varepsilon^i}{i!}\gamma_{yi} \quad \ldots\ldots\ldots \quad t\in I_2 \tag{22}$$

where

$$\alpha_x=(\Gamma_x+\Pi_xK_x),\quad \beta_{xi}=-n_{xi}^1+{}_xg_x^i,\quad \gamma_{xi}=-n_{xi}^2 \tag{23}$$

Similarly for y-subsystem.

IV. ILLUSTRATIVE EXAMPLE

Consider, a nonlinear coupled time-delay system,

$$\dot{x}_1=x_2+w \tag{24}$$

$$\dot{x}_2=-x_1-.8x_2-x_2(t-.1)-x_1^3+w \quad \text{with,} \quad x_1(\tau)=1+\tau$$

$$x_2(\tau)=-1-\tau \quad \ldots\ldots\ldots\ldots \quad -0.1\le\tau\le 0.0. \tag{25}$$

In order to be able to decouple the state model (24) let control w=u+ v= u+v and $x=x_1$ and $y=y_1$ then,

$$\dot{x}=f(x,u,\varepsilon a(.))=u+\varepsilon(y+v) \tag{26a}$$

$$\dot{y}=g(y,v,\varepsilon b(.))=-.8y+v+\varepsilon(u-x-x^3-y(t-.1)) \tag{26b}$$

Note theat (26) is now in the general coupled time delay systems of (4). The cost function is chosen to be quadratic with $L_1=F=\frac{1}{2}(x^2+u^2)$, $L_2=G=\frac{1}{2}(y^2+v^2)$ and $t_f=2.0$ seconds.

The result of the zeroth-order terms can be obtained from a pair of linear TPBV problems,

$$\dot{x}^{(0)}=-p^{(0)}, \quad \dot{p}^{(0)}=x^{(0)} \tag{27a}$$

$$\dot{y}^{(0)}=-.8y^{(0)}+q^{(0)}, \quad \dot{q}^{(0)}=y^{(0)}+.8q^{(0)} \tag{27b}$$

based on the solutions of the associated pair of Riccati equations similar to (10), (11) are obtained,

$$x^{(0)}(t)=-e^{-t}, \quad p^{(0)}(t)=e^{-t}, \quad u^{(0)}(t)=e^{-t} \tag{28a}$$

$$y^{(0)}(t)=e^{-1.28t}, \; q^{(0)}(t)=-.48e^{-1.28t}, \quad v^{(0)}(t)=-.48e^{-1.28t} \tag{28b}$$

The first-order adjoint vectors are,

$$g_x^1(t)=-.253e^{-t}+.44e^{-1.28t}+.336e^{-3.28t} \tag{29}$$

$$g_y^1(t)=-.532e^{-1.28t}+.88e^{-t}-.234e^{-3t} \quad \ldots\ldots \quad 0\leq t\leq 1.9 \tag{30a}$$

$$=-1.532e^{-1.28t}+.88e^{-t}-.234e^{-3t} \quad \ldots \quad 1.9\leq t\leq 2.0 \tag{30b}$$

The sub-optimal controls u and v after truncating the forcing functions after the first term are,

$$u\simeq -x+\varepsilon g_x^1(t) \tag{31a}$$

$$v\simeq -y+\varepsilon g_y^1(t) \tag{31b}$$

The state equations are then,

$$\dot{x}\simeq -x-\varepsilon g_x^1-\varepsilon^2 g_y^1 \tag{32a}$$

$$\dot{y}\simeq -1.8y-\varepsilon(2x+x^3+y(t-.1)+g_x^1+\varepsilon g_y^1 \tag{32b}$$

with initial functions (25). Figures 1, 2 and 3 show respectively the sub-optimal state control and performance of the nonlinear plant.

V. CONCLUSIONS

The concept of coupling perturbation is introduced for time-delay system in this paper. The procedure is to introduce a parameter ε in the original time-delay system so that the state equation can be decoupled into m(m ≥2) equations. For the case when ε→0, m un-retarded state equations are obtained.

For the case of two coupled subsystems, a suboptimal control as a MacLaurin's series expansion of ε was obtained. The suboptimal controller was shown to have an exact feedback portion and only the forward portion is approximated. It was asseted that sub-optimal control was obtained by non-delay sub-system computations only. The procedure outlined indicates that only one Riccati matrix equations must be solved regardless of the order of the approximation. The only extra computation required for each new approximation is the solution of the linear adjoint vector equation. The linear coupled time-delay systems can be similarly considered [9]. The numerical applications showed that the method can be easily implemented on the digital computer.

REFERENCES

1. G.L. Kharatishivili, "The Maximum Principle in the Theory of Optimal Processes with Time Lags", Dokl. Akad Nauk SSSR, vol. 136, No. 1, 1961.

2. N.N. Krasowskii, "Optimal Processes in Systems with Time Lags", Proc. 2nd IFAC Congress, Basel, Switzerland, 1963.

3. D.H. Eller, J.K. Aggarwal and H.T. Banks, "Optimal Control of Liniar Time-Delay Systems", IEEE Trans. AC., vol. AC-14, No. 6, December, 1969, pp. 678-687.

4. J.K. Aggarwal, "Computation of Optimal Control for Time-Delay Systems", IEEE Trans. AC., vol. AC-15, No. 6, December, 1970, pp. 683-685.

5. K. Inone, H. Akashi, K. Ogino and Y. Sawaragi "Sensitivity Approaches to Optimization of Linear Systems with Time-Delay", IFAC Automatica, vol. 7, No. 6, November, 1971, pp. 671-679.

6. P. Sannuti, "Near Optimum Design of Time-Lag Systems by Singular Perturbation Method", Proc. 1970 JACC, Atlanta, GA., pp. 489-496.

7. P. Sannuti and P.V. Kokotovic', "Near Optimum Design of Linear Systems by Singular Perturbation Method", IEEE Trans. AC., vol. AC-14, No. 1, February, 1969, pp. 15-22.

8. P.V. Kokotovic', and G. Singh, "Optimization of Coupled Nonlinear Systems", Int. J. of Control, vol. 14, No. 1, 1971, pp. 51-64.

9. M. Jamshidi, "Suboptimal Control of Coupled Time Delay Systems", to appear in Int. J. Control.

10. R.E. Kalman, "Contribution to the Theory of Optimal Control", Bol. Soc. Math. Mexico, 2nd Series, vol 5, 1960, pp. 102-118.

APPENDIX

TPBV = two-point boundary value
ε = coupling parameter, scalar
λ = time delay
t_o = initial process time
t_f = final process time
T = transposition of vectors and matrices
$x' = x(t-\lambda)$
$x(i) = \lim_{\varepsilon\to 0} \partial x/\partial\varepsilon^i$
$\dot{x} = dx/dt$
$s = t+\lambda$

The expressions for the "sub-system x" are,

$$A_x=f_x-f_u(f^2-F_{uu})^{-1}(f_x^2-F_{ux})$$

$$S_x=f_u(f_u^2-F_{uu})^{-1}f_u^T$$

$$Q_x=(F_{xx}-f_x^1)(F_{xu}-f_u^1)(f_u^2-F_{uu})^{-1}(f_x^2-F_{uu})$$

$$\Gamma_x=(f_u^2-F_{uu})^{-1}(f_x^2-F_{ux})$$

$$\Pi_x=(f_u^2-F_{uu})^{-1}f_u^T$$

$$L_x^1=f_\alpha a-f_u(f_u^2-F_{uu})^{-1}U_1^1\ ,\quad L_x^2=f_\alpha a-f_u(f_u^2-F_{uu})^{-1}U_2^1$$

$$\ell_x^1=Q_c-(F_{xu}-f_u^1)(f_u^2-F_{uu})^{-1}U_1^1$$

$$\ell_x^2=Q_c-(F_{xu}-f_u^1)(f_u^2-F_{uu})^{-1}U_2^1$$

$$\eta_x^1=(f_u^2-F_{uu})^{-1}U_1^1$$

$$\eta_x^2=(f_u^2-F_{uu})^{-1}U_2^1$$

where $Q_c \equiv -(f_\alpha^1 a+a_x^T f_\alpha^T p+b_x^T g_\beta^T q)$.

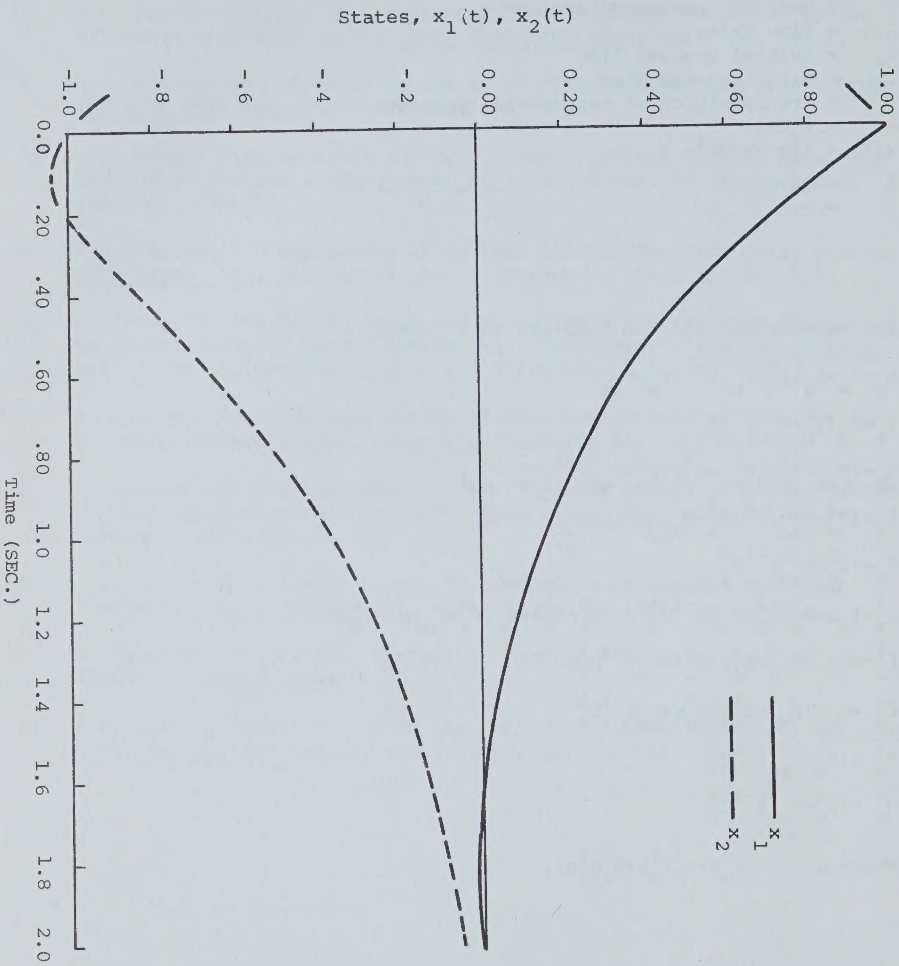


Fig. 1 Time responses for the states x_1 and x_2

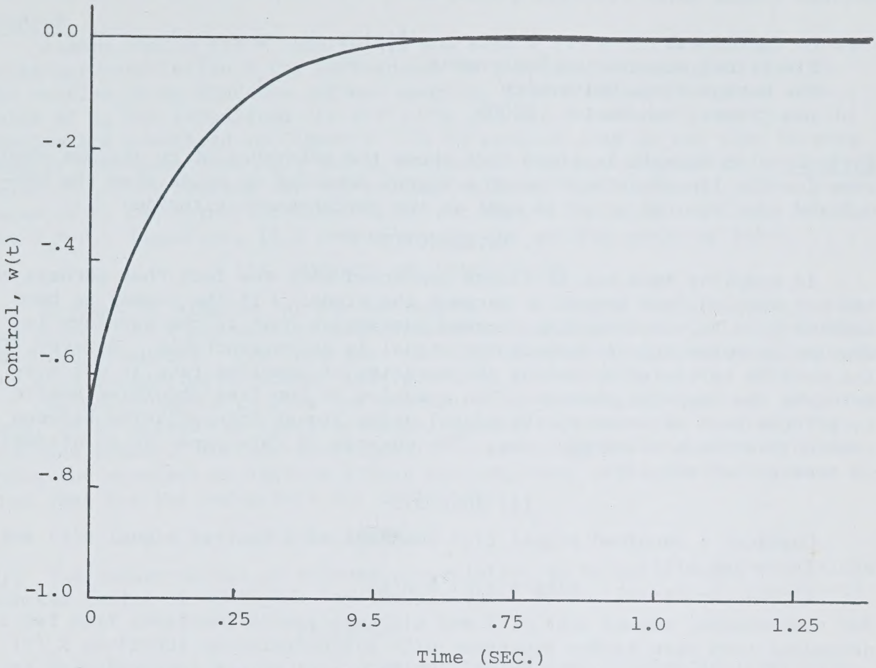


Fig. 2 Control function vs. time

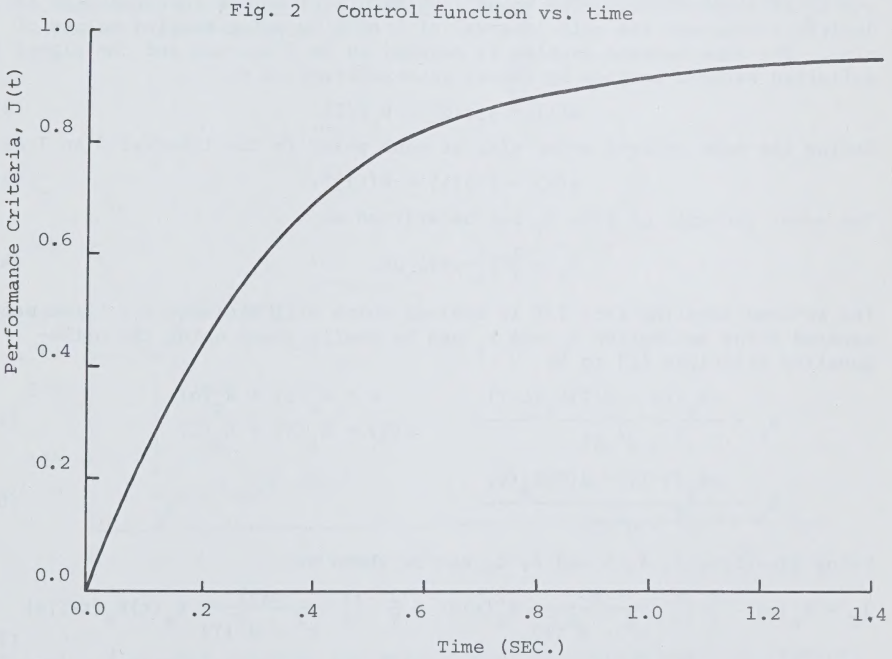


Figure 3. Cost Criteria vs. Time

OPTIMUM SAMPLE RATES FOR DATA ESTIMATION

L. C. Ludeman
Electrical Engineering Department
New Mexico State University
Las Cruces, New Mexico 88003

Abstract - An example is given that shows the existence of an optimum sample rate for the linear estimation of a signal imbedded in noise when the normalized mean squared error is used as the performance criterion.

I. INTRODUCTION

In sampling data one is always concerned with the fact that perhaps he has not sampled fast enough to recover the signal. If the signal is band limited to W Hz, the sampling theorem guarantees that if the waveform is sampled at twice this frequency the signal is reconstructable. However, if the data is corrupted by noise, the question of sampling rate is not answered by the sampling theorem. The question of how fast should we sample to get the best estimate of the signal using linear interpolation between samples becomes a meaningful one. The purpose of this paper is to attempt to answer that question.

II. ANALYSIS

Consider a received signal r(t) composed of a desired signal s(t) and additive noise n(t).

$$r(t) = s(t) + n(t) \tag{1}$$

For convenience, assume that s(t) and n(t) are sample functions from two independent zero mean random processes with autocorrelation functions $R_s(\tau)$ and $R_n(\tau)$ respectively. The "best" estimate $\hat{s}(t)$ of the received s(t) is desired throughout the unit interval of 0 to 1 by using sampled values of r(t). The time between samples is assumed to be T seconds and the signal is estimated between samples by linear interpolation, i.e.

$$\hat{s}(t) = a_t r(o) + b_t r(T) \tag{2}$$

Define the mean squared error e(t) at each point in the interval 0 to T as

$$e(t) = E\{s(t) - \hat{s}(t)\}^2. \tag{3}$$

The error per unit of time I_T can be written as

$$I_T = \frac{1}{T}\int_o^T e(t)\,dt. \tag{4}$$

The optimum sampling rate 1/T is desired which will minimize I_T. From mean squared error estimation a_t and b_t can be easily found using the orthogonality principle [1] to be

$$a_t = \frac{cR_s(t) - d(T)R_s(t-T)}{c^2 - d^2(T)}, \qquad \begin{aligned} c &= R_s(o) + R_n(o) \\ d(T) &= R_s(T) + R_n(T) \end{aligned} \tag{5}$$

$$b_t = \frac{cR_s(t-T) - d(T)R_s(t)}{c^2 - d^2(T)} \tag{6}$$

Using equations 3, 4, 5 and 6, I_T can be shown as

$$I_T = R_s(o) - \frac{2}{T}\int_o^T \frac{c}{c^2 - d^2(T)} R_s^2(t)dt + \frac{2}{T}\int_o^T \frac{d(T)}{c^2 - d^2(T)} R_s(t)R_s(t-T)dt \tag{7}$$

The value of T for which I_T is minimized gives the desired time between samples for data estimation. The existence of such an optimum sample rate is shown in the following example.

EXAMPLE

Assume that $R_s(\tau) = \exp\{-|\tau|\}$ and that $R_n(\tau) = \text{Sin } 2\pi WT/2\pi WT$. This noise autocorrelation $R_n(\tau)$ corresponds to a band limited signal at W Hz. The problem is to find the optimum sampling time T between samples. The value of I_T was calculated for W = 10 Hz and all T between 0 and 1 and the results are presented in figure 1. It is noticed that as the time between samples decreases from one, the I_T decreases (not monotonically) until a value of .29 is reached for the related value of T = .069 sec. After that value of T, the error increases until the unavoidable error of .5 is found for T = 0. Therefore, 14.5 sample/sec is the optimum sampling rate.

III. SUMMARY AND CONCLUSIONS

A specific example for the estimation of a sample function from a random process with exponential autocorrelation imbedded in band limited noise has been presented. For this case the existence of an optimum sampling rate for data estimation by linear interpolation between samples was given.

Although useful to show the procedure for finding the optimum sampling rate, if it exists for specific cases, general behavior and existence has not been proved. Further investigation into the general behavior by modifying the approach to include linear and nonlinear interpolation by more than just the two end-points may be fruitful.

ACKNOWLEDGEMENT

The author wishes to express appreciation to Gilberto Mares and Bertram Leslie of MEWTA/EWL for support under Grant #DA-AROD-31-124-72-G53.

REFERENCE

[1] Papoulis, A., Probability Random Variables and Stochastic Processes, McGraw-Hill, New York, 1965.

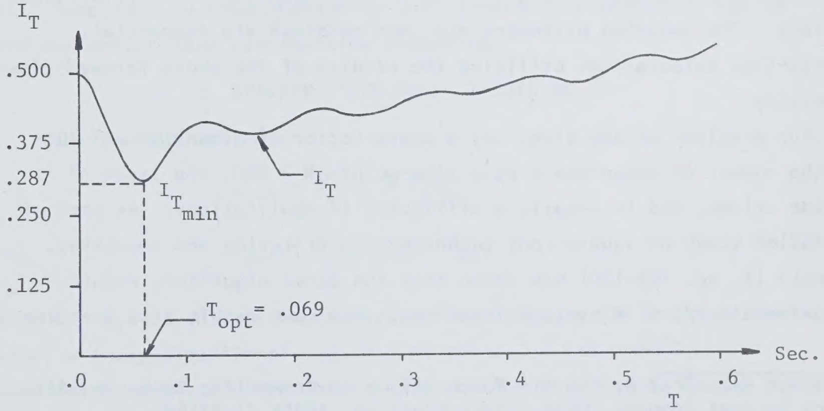


Fig. 1. Plot of I_T showing the existence of an optimum sampling interval.

ON THE DATA STORAGE PROBLEM IN SEQUENTIAL DISCRETE-TIME SMOOTHING*

J.S. MEDITCH
University of California
Irvine, California 92664

ABSTRACT

Data storage requirements for fixed-interval (off-line) smoothing for discrete-time, linear systems are reviewed. These are compared with those for two on-line procedures, fixed-point and fixed-lag, under the assumption that all algorithms process the same measurement data. Explicit conditions under which one formulation requires less storage than the others are developed.

I. INTRODUCTION

Two well-known algorithms for sequential, optimal smoothing for discrete-time, linear systems are due to Bryson and Frazier [1, pp.393-394], and Rauch, Tung, and Striebel [1, p. 393; 2, pp. 224-225]. The two are algebraically equivalent in that one is easily derivable from the other. Both are of the fixed-interval type [2, pp. 204-208], i.e., they are off-line (post-experimental) procedures for obtaining the optimal (least-squares, minimum variance, etc.) state estimate at each time-point k within a fixed measurement data span [0, N] where the estimate is based on all the measurements in [0, N]. Both require first a forward-time pass through the measurement data to obtain the optimal predicted and filtered state estimates along with their corresponding error covariance matrices. The smoothed estimates are then obtained via sequential, reverse-time calculations utilizing the results of the above forward-time processing.

For problems of any size, say a state vector of dimension $n \geqslant 10$ and the number of measurement data time-points $N \geqslant 500$, the issue of storage arises, and is clearly a difficulty in applications. As part of a detailed study on square-root techniques in filtering and smoothing, Kaminski [3, pp. 104-105] has shown that the above algorithms require approximately $1/2\ n^2\ N$ storage locations. The same result also pertains

---

*Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR 71-2116A.

to the Fraser [4] "two-filter" smoothing algorithm and its square-root version. This latter formulation involves simultaneous operation of two filters one of which sweeps forward in time from 0 to $k \varepsilon (0, N)$ and the other which sweeps from N to k in reverse-time. For $n = 20$ and $N = 1000$, approximately 200,000 storage locations are thus needed in these formulations.

A considerable reduction in storage requirements can be effected by utilizing the so-called square-root information smoother of Kaminski [3, pp. 85-91]. Here the forward sweep is made by the square-root information filter developed by Dyer and McReynolds [5]. The required number of storage locations is of the order npN where p is the dimension of the dynamic system noise disturbance. Thus, again for a system with $n = 20$, $N = 1000$ and say $p = 5$, the number of storage locations is approximately 100,000, one-half that above. The factor becomes one-fourth if it is found that double precision is needed in the former algorithms.

In this paper, two other smoothing formulations for processing the same measurement data as above, i.e., that over [0, N], are examined. The motivation for the study is two-fold. First, fixed-interval smoothing is inherently an off-line procedure whereas the "stacked" fixed-point and the fixed-lag formulations to be considered here can be implemented on-line (forward-time) along with the Kalman filter to perform sequential prediction, filtering, and smoothing simultaneously. The purpose then is to determine what, if any, the additional cost in storage is for on-line operation. Second, even if on-line processing is not required, it is still important to know whether or not forward-time smoothing can be more economical than reverse-time smoothing.

2. STACKED FIXED-POINT SMOOTHING

Consider the system

$$x_{k+1} = \Phi_k x_k + \Gamma_k w_k \tag{1}$$

$$z_{k+1} = H_{k+1} x_{k+1} + v_{k+1} \tag{2}$$

where $k = 0,1,\ldots,N-1$ is the discrete-time index, x the n-dimensional state, w the p-dimensional system disturbance, z the m-dimensional measurement, v the m-dimensional measurement error, and Φ, Γ, and H are known nxn, nxp, and mxn matrices, respectively. Let x_o be a zero mean, gaussian, random vector with covariance P_o, and $\{w_k\}$ and $\{v_{k+1}\}$ be zero mean, gaussian, white sequences with covariance matrices Q_k and R_{k+1}, respectively. Assume also that x_o, $\{w_k\}$, and $\{v_{k+1}\}$ are mutually independent.

Let $\hat{x}_{k|j}$ denote the optimal estimate of x_k based on all measurements up to and including the one at time j. The corresponding estimation error is defined by $\tilde{x}_{k|j} = x_k - \hat{x}_{k|j}$, and the associated error covariance is denoted $P_{k|j}$. Prediction is defined for $j = k-1$, filtering for $j = k$, fixed-interval smoothing for $j = N$ and $k < N$, fixed-point smoothing for k fixed and $j = k+1, k+2, \ldots, N$, and fixed-lag smoothing for $k = j - L$ where L is a positive integer, the lag, and $j = L, L+1, \ldots, N$.

Since smoothing generally depends on prediction and filtering for input data, the Kalman filter equations are presented here for reference purposes:

$$\hat{x}_{k+1|k} = \Phi_k \hat{x}_{k|k}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \tilde{z}_{k+1|k}$$

$$\tilde{z}_{k+1|k} = z_{k+1} - H_{k+1} \hat{x}_{k+1|k}$$

$$K_{k+1} = P_{k+1|k} H'_{k+1} (H_{k+1} P_{k+1|k} H'_{k+1} + R_{k+1})^{-1}$$

$$P_{k+1|k} = \Phi_k P_{k|k} \Phi'_k + \Gamma_k Q_k \Gamma'_k$$

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}$$

where $k = 0, ., \ldots, N-1$, prime denotes the transpose, $(\quad)^{-1}$ the matrix inverse, I the identity matrix, and K the Kalman filter gain matrix. Further, $\hat{x}_{o|o} = 0$ and $P_{o|o} = P_o$.

Turning now to the fixed-point smoothing problem a convenient forward-time, sequential algorithm for this purpose is [2,pp. 232-233]:

$$\hat{x}_{j|k} = \hat{x}_{j|k-1} + W_{j|k} H'_k R_k^{-1} \tilde{z}_{k|k-1} \tag{3}$$

$$W_{j|k} = W_{j|k-1} \Phi'_k (I - K_k H_k)' \tag{4}$$

$$P_{j|k} = P_{j|k-1} - W_{j|k} (S_k P_{k|k-1} S_k + S_k) W_{j|k} \tag{5}$$

$$S_k = H'_k R_k^{-1} H_k$$

for any fixed $j \varepsilon [0, N)$ with $k = j+1, j+2, \ldots, N$. Here, W is the n x n gain matrix, and all other terms have been defined previously. It should be noted that the measurement error covariance matrix R_k is assumed positive-definite for all relevant k, and that unlike the situation in filtering, the gain calculation, Eq. (4), is independent of the estimation error covariance computation, Eq. (5). The initial conditions for Eqs. (3), (4), and (5) are $\hat{x}_{j|j}$, $P_{j|j}$, and $P_{j|j}$, respectively. The manner in which the algorithm of Eqs. (3) and (4) operates in parallel with the Kalman filter should be clear.

The set of estimates obtained from fixed-interval smoothing for the system defined by Eqs. (1) and (2) is $\{\hat{x}_{N-1|N}, \hat{x}_{N-2|N}, \ldots, \hat{x}_{1|N}, \hat{x}_{0|N}\}$ generated in the order given. Reference to Eq. (3) for $j = 0$ and $k = 1,2,\ldots,N$ reveals that fixed-point smoothing yields the set $\{\hat{x}_{0|1}, \hat{x}_{0|2}, \ldots, \hat{x}_{0|N}\}$ also calculated in the indicated order. Of this latter set, only $\hat{x}_{0|N}$ is in the former set. To obtain the other elements in the fixed-interval set via fixed-point smoothing, additional fixed-point smoothers must be implemented at successive values of j, $j=1,2,\ldots,N-1$, where, for each j, the index for the forward sweep is $k = j+1, j+2,\ldots,N$. Thus, with $j = 1$, the set $\{\hat{x}_{1|2}, \hat{x}_{1|3}, \ldots, \hat{x}_{1|N}\}$ evolves, while for $j = N-1$, the set $\{\hat{x}_{N-1|N}\}$ is obtained. This leads to the notion of "stacked" fixed-point smoothing wherein an additional fixed-point smoother comes on-line at each measurement time. The complete set of smoothed state estimates that results in this way is shown below with the corresponding subset for fixed-interval smoothing alone enclosed in the dashed lines.

$$\begin{array}{ccccccc|c|}
\cline{8-8}
\hat{x}_{0|1} & \hat{x}_{0|2} & \hat{x}_{0|3} & \cdot & \cdot & \hat{x}_{0|N-1} & & \hat{x}_{0|N} \\
 & \hat{x}_{1|2} & \hat{x}_{1|3} & \cdot & \cdot & \hat{x}_{1|N-1} & & \hat{x}_{1|N} \\
 & & \hat{x}_{2|3} & \cdot & \cdot & \hat{x}_{2|N-1} & & \hat{x}_{2|N} \\
 & & & \cdot & \cdot & \cdot & & \cdot \\
 & & & & & \hat{x}_{N-2|N-1} & & \hat{x}_{N-2|N} \\
 & & & & & & & \hat{x}_{N-1|N} \\
\cline{8-8}
\end{array}$$

An obvious disadvantage of this formulation is the expanding storage requirement which reaches its maximum at time N-1. To get a comparison with Kaminski's results, the same assumptions will be applied: (1) storage of Φ_k, Γ_k, H_k, Q_k and R_k is not considered, and (2) symmetric matrices are stored in upper (lower) triangular form.

The Kalman filter which operates concurrently with the smoother requires storage of $\hat{x}_{k|k}$ and $P_{k|k}$ at each stage for a total of n(n+3)/2 locations. Reference to the above table and Eqs. (3) and (4) shows that the stacked fixed-point smoother needs, at k = N-1, to store $\{\hat{x}_{0|N-1}, \hat{x}_{1|N-1}, \ldots, \hat{x}_{N-2|N-1}\}$ and $\{W_{0|N-1}, W_{1|N-1}, \ldots, W_{N-2|N-1}\}$ which amounts to n(n+1)(N-1) locations. The total is, therefore,

$$ST = \frac{n(n+3)}{2} + n(n+1)\ (N-1)$$

which for sufficiently large n and N is $ST \cong n^2N$. This is twice the storage requirement for the conventional fixed-interval formulations. The same factor applies in comparing the square-root fixed-interval and "stacked" fixed-point forms. Considering no other factors, this is apparently the price that must be paid if one wishes to employ this on-line smoothing formulation. On the other hand, there exist certain features and options within the current approach that merit consideration. First, it should be noted that the smoothers in the stack operate independently of each other. Thus, any subset of them can be deleted without affecting the others. If, for example, it is found that knowledge of only one-half the elements in the fixed-interval set are of interest, the storage requirement for the fixed-point smoothing is halved. Conventional fixed-interval smoothing cannot accomplish the same task since its natural sequence of calculations is $\hat{x}_{N-1|N} \rightarrow \hat{x}_{N-2|N} \rightarrow \hat{x}_{N-3|N} \rightarrow \cdots \rightarrow \hat{x}_{0|N}$, none of which can be skipped.

Second, fixed-interval smoothing yields N smoothed state-vector estimates while fixed point gives N(N+1)/2 of them. In practical applications, the availability of these intermediate smoothed estimates may give the engineer greater insight into the dynamic system being analyzed. Contrary to this may be the fact that generation of the entire set of smoothed estimates is not necessary. For example, in a problem where N = 1000, it may be found that $\hat{x}_{0|100}$ is as "good" an estimate of x_0 in the sense of smoothing error covariance as $\hat{x}_{0|1000}$. In this case, the first smoother in the stack ceases operation at N = 100. This is in contrast to the fixed-interval formulations which can only obtain a smoothed estimate of x_0 by calculating $\hat{x}_{0|1000}$. The same type of argument may apply to other smoothed estimates. Thus, the possibility exists that stacked fixed-point smoothing may actually require less storage than fixed-interval smoothing in order to obtain basically the same set of state estimates from the given data. The extent to which this is possible can be determined by implementing Eqs. (4) and (5) to perform an error covariance analysis. Storage is not a problem in this case since the error analysis can be conducted separately for each value of j, $j = 0,.,\ldots,N-1$, if need be. In cases where the Kalman filter is exponentially, asympototically stable, it has been shown [6] that all of the effective estimation error covariance reduction that smoothing offers occurs by smoothing over an interval of three to four times the largest time constant of the filter. This proves to be an important point as well for the fixed-lag formulation in the sequel.

Finally, it is remarked that the use of stacked square-root fixed-point smoothing can be employed in any of the above cases to effect a further storage requirement savings. In the worst case, one has $ST = 2npN$. With p the order of n/4, which is not unreasonable, $ST = 1/2\ n^2 N$. Under these conditions, the storage, while still severe, is not prohibitive for a wide range of problems.

3. FIXED-LAG SMOOTHING

The use of fixed-lag smoothing as a scheme for reducing the storage requirement in solving the smoothing problem has been suggested by Kaminski [3, p. 106]. This coupled with the above remark on the effective interval for smoothing [6] will show that there do exist situations where fixed-lag smoothing is attractive.

Fixed-lag smoothing algorithms which avoid an internal stability problem found in previous such algorithms [7, 8] have been proposed in [9-11]. Letting L denote the lag, the estimates sought are $\hat{x}_{j-L|j}$ for $j = L, L+1,\ldots,N$. By augmenting the state vector of Eq. (1) to include $x_k, x_{k-1},\ldots,x_{k-L}$, i.e., delays of the state, it has been shown that the optimal fixed-lag smoother can be viewed as an nL-dimensional Kalman filter with delays. At each value of j, the smoother gives the entire set of smoothed estimates in [j-L, j), i.e., $\{\hat{x}_{j-L|j}, \hat{x}_{j-L+1|j},\ldots,\hat{x}_{j-1|j}\}$ and, of course, the filtered estimate $\hat{x}_{j|j}$. The same set of estimates could be generated with fixed-interval smoothing over [j-L, j], but without the option of having j index recursively in forward-time.

Since the fixed-lag smoother is merely a Kalman filter, its storage requirement, subject to the previously stated assumptions, is

$$ST = \frac{n(n + 3)}{2} L$$

For $n \gg 3$, this gives $ST \cong 1/2\ n^2 L$.

It is now possible to make two important comparisons of storage requirements. First, consider one of the conventional fixed-interval smoothers and assume that the fixed-lag smoother is implemented using the conventional Kalman filter equations. Then, in order for fixed-lag smoothing to require less storage than fixed-interval smoothing, it is necessary that

$$1/2\ n^2 L < 1/2\ n^2 N$$

or $L < N$.

As a second comparison, suppose that Kaminski's square-root information smoother is used and that square-root filtering is employed to realize the fixed-lag smoother. Storage for the square-root filter is approximately the same as that for the conventional filter. Thus, in order to favor fixed-lag smoothing, one must have

$$1/2 n^2 L < n\ p\ N$$

or $L < \frac{2\ p\ N}{n}$. For a previous example where $n = 20$, $p = 5$ and $N = 1000$, $L < 500$ is needed.

The square-root algorithms are, of course, preferable because of their numerical properties [3].

4. CONCLUSION

Storage requirements for fixed-interval (off-line) smoothing algorithms have been compared with those for two on-line procedures, fixed-point and fixed-lag, under the assumption that all algorithms are to process the same set of measurement data. It was found that the fixed-point algorithm generally requires approximately twice as much storage as the fixed-interval method, but possesses options, depending on error analysis results, that can demand even less storage than the latter. In the case of fixed-lag smoothing, explicit conditions on the amount of lag were developed for both conventional and square-root formulations in order to achieve a smaller storage requirement.

The question of the associated computational complexity has not been treated here, in part since the storage problem appears at present to be the more critical one. However, this issue must certainly be resolved before a viable comparison between off-line and on-line smoothing procedures can be claimed.

REFERENCES

1. A. E. Bryson, Jr. and Y. C. Ho, "Applied Optimal Control," Blaisdell, Waltham, Mass., 1969.

2. J.S. Meditch, "Stochastic Optimal Linear Estimation and Control," McGraw-Hill, New York, 1969.

3. P.G. Kaminski, "Square Root Filtering and Smoothing for Discrete Processes," SUDAAR NO. 427, Center for Systems Research, Guidance and Control Laboratory, Stanford Univ., Stanford, Calif., July, 1971.

4. D.C. Fraser, "A New Technique for the Optimal Smoothing of Data," T-474, Instrumentation Laboratory, Mass. Inst. of Technology, Cambridge, Mass., January 1967.

5. P. Dyer and S.R. McReynolds, "Extension of Square-Root Filtering to Include Process Noise," Jour. Optimization Th. and Appl., vol. 3, No. 6, 1969, pp. 444-459.

6. B.D.O. Anderson, "Properties of Optimal Linear Smoothing," IEEE Trans. Auto. Control, vol. AC-14, no. 1, February 1969, pp. 114-115.

7. C.N. Kelly and B.D.O. Anderson, "On the Stability of Fixed-Lag Smoothing Algorithms," Jour. Franklin Inst., vol. 291, no. 4, April 1971, pp. 271-281.

8. J.S. Meditch, "Stability in Fixed-Lag Smoothing," Proc. 1972 Joint Auto. Control Conf., Stanford, Calif., August 1972, pp. 838-841.

9. K.K. Biswas and A.K. Mahalanabis, "Application of State Augmentation Approach to Problems of Optimal Smoothing," Dept. of Elec. Engr., Indian Inst. of Tech., New Delhi, India, February 1971.

10. J.B. Moore, "Discrete-Time Fixed-Lag Smoothing Algorithms," Automatica, March 1973, to appear.

11. K.K. Biswas and A.K. Mahalanabis, "A Stable Fixed-Lag Smoothing Algorithm," to appear.

INTERMEDIATE RANGE PEAK LOAD DEMAND FORECASTING

K.R. SHAH,* B.V. BALIGA and Y. YAVIN**
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

ABSTRACT

This paper discusses a hybrid technique consisting of a combination of the Kalman filter state estimation technique and a modified exponential smoothing or ratio matrix technique (for penalizing state variable, Δs, representing system characteristic) for forecasting intermediate range peak load demands in electric utility systems.

INTRODUCTION

There are three broad categories of forecasts; namely, long term, short term, and intermediate range, for meeting the future electrical load demands reliably and economically. Long term forecast ranging from 5 to 10 years is useful to load forecaster in planning major generation and transmission facilities. Any forecasting technique for predicting load demands as much as ten years in advance, is liable to be in error because of the uncertainty of socio-economic factors; however, Delphi forecast[1, 2] promises better results. Short term forecasts for predicting weekly, daily and hourly loads is useful for reliable and economical system operation and maintenance and methods of predicting this type of load demand can be found in Refs. 3-6. Intermediate range forecast, on the other hand, considers prediction of loads one to three years in advance. The knowledge of this type of load demand is useful in examining the existing and planned generating capacity and ordering of peaking units if generation deficiency is ascertained. Hence, accurate prediction of load demands in intermediate range could help prevent power brown-outs which are frequent during summer months. The importance of a better technique for intermediate range forecasting lies in the latter fact.

There are four major aspects of any forecasting study: (i) the availability of raw data, (ii) formulation of model to correspond to the physical problem in question, (iii) selection of a forecasting method, and (iv) its implementation to physical (real world) problems. Most of the major utilities have instrumentation and data-logging equipment for metering and reading load history. Hence, this paper is concerned with the remaining aspects of the forecasting problem.

---

*Presently associated with Hughes Research Laboratories, Malibu, CA

**Currently working with the University of The Negev, Israel.

This paper proposes (i) a better state model with state variables $X(k) = \text{col.} \left[x(k)\ \Delta w(k)\ \Delta gd(k)\ \Delta s(k) \right]$ and (ii) a new hybrid technique consisting of a combination of the Kalman filter state estimation technique and a modified exponential smoothing or a ratio matrix technique (for penalizing state variables $-\Delta s-$) for forecasting intermediate range load demands. The results thus obtained give a maximum error of 3.8% and 4.5% for the first and second year predictions, respectively.

FORMULATION OF A STATE MODEL

The state model is constructed by noting that the observed load demand $y(k)$ at instant k differs from the actual demand $X(k)$ by the noise component $\eta(k)$ consisting of instrumentation noise. The estimated value of load demand $X(k+1)$ at $(k+1)$st instant is then equal to the system load demand $X(k)$ at kth instant plus the incremental load demands contributed by (i) weather conditions - $\Delta w(k)$ - with associated noise component $\xi_w(k)$, (ii) changes in system behavior or characteristic $\Delta s(k)$, and (iii) changes related to socio-economic factors - Δgd - with associated noise component $\xi_{gd}(k)$. We assume the noises are white and independent of each other.

Thus, a state model representing intermediate range load demand as defined by equations (1) to (3) is given by

$$X(k+1) = \Phi(k)\, X(k) + \Gamma(k)\, \Sigma(k) \tag{1}$$

$$y(k) = H(k)\, X(k) + \eta(k) \tag{2}$$

where

$$X(k) = \text{Col.} \left\{ x(k),\ \Delta w(k),\ \Delta gd(k),\ \Delta s(k) \right\}$$

$$\Sigma(k) = \text{Col.} \left\{ \xi_x(k),\ \xi_w(k),\ \xi_{gd}(k),\ 0 \right\}$$

$$\Phi(k) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_k \end{bmatrix}$$

$$a_k = 1 + \frac{\Delta s(k-1) + \Delta s(k-2)}{2\Delta s(k)}$$

$$H(k) = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\Gamma(k) = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \tag{3}$$

In state equations (1) to (3) we define $x(k)$ as the system state, $y(k)$ the observation, $\Sigma(k)$ as the noise vector and $\eta(k)$ as the observation noise.

IMPLEMENTATION USING KALMAN FILTERING TECHNIQUE

The purpose of deriving the suitable state model was to obtain $X(k+1/k)$ with the observations $y(k)$ at k given. The output of Kalman filter would be the solution for the best estimate $X(k+1)$ given by Ref. 7

$$\hat{X}(k+1/k) = \Phi(k)\,\hat{X}(k/k-1) + K(k)\left\{y(k) - H(k)\,\hat{X}(k/k-1)\right\} + \Gamma(k)\,\bar{\Sigma}(k) \tag{4}$$

where $K(k)$ is the Kalman gain.

A general purpose program - Linear Quadratic Loss - developed by Bullock and Fosha[8] was used to obtain the final estimate of the state at $k+1$.

The computer implementation procedure for forecasting the intermediate load demand as defined by a state model (state equations (1) to (3)) in using the Kalman filtering technique is given in Fig. 1. However, in order to start the Kalman filter program, initial values of the states defined above are needed.

Determination of Initial Values

The observed (raw) data were preprocessed to obtain time independent, weather independent and time and weather independent data as shown in Fig. 2.

Time-regression and weather-regression analyses are conducted to find initial values $\Delta w(o)$ and $\Delta gd(o)$. Since we know $y(o)$ and $\Delta gd(o)$ and assuming the initial error of 5% between $y(o)$ and state $x(o)$ we can evaluate the initial system behavior $\Delta s(o)$. These quantities $\{x(o)\ \Delta w(o)\ \Delta gd(o)\ \Delta s(o)\}$ are essential to initiate the Kalman filter program. We note that regression analysis is used only to obtain the initial values and is not conducted at every stage of prediction.

Implementation Procedure

The computer implementation procedure (shown in block diagram form in Fig. 1) for forecasting intermediate-range load demand using Kalman filtering technique is described below:

(i) The first k observation values are given as data $(y_1, y_2, \ldots, y_k)$ and the weather parameters are incorporated as state Δw and Δgd.

(ii) The regression is used to find the initial states Δw_1 and Δgd_1. Knowing these we note from the eq. (1) that Δs_1 is still unknown.

(iii) In order to determine Δs_1 we will assume an initial error of 5% between the observation y_1 and the state x_1 and obtain Δs_1 from eq. (2). We will also assume an initial α_1 of unity i.e. the behavior of the system remains same initially.

(iv) These values are fed into the Kalman filter program[8] which has built in random generator. The mean of the noises ξ_x, ξ_w and ξ_{gd} is assumed to be 1% of their respective states. η has a mean zero.

(v) The output of the Kalman filter is $\hat{X}_2$ (i.e. Col. $\{x_2, \Delta\omega_2, \Delta gd_2, \Delta s_2\}$) given by the equation (4) as

$$\hat{X}(2/1) = \phi(1)x(1) + K(1)\ y(1) - H(1)X(1) + \Gamma(1)\bar{\Sigma}(1)$$

(vi) The regression can now be used to check any abnormality in the weather state $\Delta\omega_2$ or GNP and DD state Δgd_2.

(vii) Δs_2 is used to obtain a new assessment of α which will be called α_2.

(viii) The necessary changes are made in the vector $\hat{X}_2$. (Note that Σ in the block diagram indicates the collection of all the necessary parameters rather than their algebraic sum). The new $\hat{X}_2$ and the modified coefficient matrix in eq. (1) is fed into the Kalman filter along with the observation. Again the noise is assumed as in (iv).

(ix) The output of the filter is $\hat{X}_3$ from eq. (4). The output of the filter also includes the variance vector $P(k + 1)$ at every stage.

(x) The procedure is recurrent from now on, i.e., the step (ii) to (ix) are repeated.

(xi) The penalty factor for years from 1960 to 65 on Δs is assumed unity. Later on a new procedure is evaluated to insert penalty factor on Δs.

(xii) The procedure is repeated till we get desired $\hat{X}(k)$.

DISCUSSION OF RESULTS USING KALMAN FILTER

The actual data from Baltimore Gas and Electric Company is available from 1960-70. However, observations from 1960-67 were used to predict 1968 and 1969 states and compared with actual values. The results thus obtained are plotted in Tables I and II. From Table I it is seen that the percent error for yearly prediction (i.e. 1968 forecast) was mostly within 6% and the maximum error was -6.37%.

Such a high percentage error for yearly forecast using Kalman filter technique cannot be tolerated. Hence a need to reduce this error was apparent. When one of the states - Δs - representing system characteristic was plotted and least square line was fitted, the approximated curve of Fig. 3 gave significance regarding the energy required by the system. We note that during the summer months the area ander the curve is large. This is the energy that the power company has failed to take into account due to changes in system behavior. In winter months, however, there is a reserve energy left over which was not required by the system according to its behavior. This led us to the idea that state Δs needs to be modified by developing a penalty factor to be imposed on the system. The weighting procedure to estimate such a penalty factor is discussed next.

TABLE I

RESULTS FOR YEAR 1968

Year 68 Month	Actual	Kalman Filter		Hybrid Model	
		Prediction	Percent Error	Prediction	Percent Error
January	1509	1543	2.24	1537	1.86
February	1478	1421	-3.86	1444	-2.3
March	1421	1473	2.52	1469	3.8
April	1370	1400	2.14	1378	0.58
May	1387	1312	-5.7	1346	-3.06
June	1903	1876	-1.43	1871	-1.71
July	2023	1905	-6.37	1958	-3.2
August	2058	1968	-3.3	2032	-1.26
September	1720	1624	-5.86	1658	-3.6
October	1486*	1663	----	1684	----
November	1557	1637	4.95	1612	3.52
December	1631	1638	0.42	1638	0.42

Note: All loads are in MW
*Outliers
+ error indicates optimistic forecast
- error indicates pessimistic forecast

TABLE II

RESULTS OF YEAR 1969

Year 69 Month	Actual	Adjusted by Company	Kalman Filter		Hybrid Model	
			Prediction	Percent Error	Prediction	Percent Error
January	1637	1637	1663	1.59	1667	1.83
February	1599	1599	1632	2.06	1628	1.81
March	1370*	1600	1549	-3.14	1555	-2.81
April	1486	1486	1518	2.16	1516	2.02
May	1781*	1794	1479	----	1488	----
June	2111	2121	2143	1.03	2134	0.61
July	2169	2110	2089	-3.63	2103	-2.92
August	2095	2095	1993	-4.86	2006	-4.26
September	2082*	2082	1891	----	1935	----
October	1617	1617	1692	5.24	1684	4.47
November	1660	1660	1699	2.34	1702	2.53
December	1768	1768	1693	-3.78	1715	3.00

Penalty Factor Algorithm

A complete procedure to get penalty factor is given below:

1. Use values M_1, M_2, ..., M_5
2. Use e^k in eq. (5) to get δ.

$$K = \ell n \frac{x - M_5 \gamma}{M_5 - M_4}, \quad Y = e^{\alpha} + e^{\beta} + e^{\gamma} \qquad (5)$$

3. If δ is obtained and satisfies $\alpha < \beta < \gamma < \delta$ then proceed to use the penalty factor so obtained.

4. But if $\delta < \gamma$ or δ cannot be evaluate, use average matrix

$$\begin{bmatrix} \frac{a_1 + a_2}{2} & 0 \\ 0 & \frac{b_1 + b_2}{2} \end{bmatrix}$$

to obtain the penalty factor.

PREDICTION BY HYBRID METHOD AND DISCUSSION OF RESULTS

Hybrid method as defined here is a combination of Kalman filtering technique and penalty factor algorithm. Penalty factor derived in the previous section is used to change Δs to Δs_n. Then the vector would be

$$X(k) = \left\{ x(k)\ \Delta w(k)\ \Delta gd(k)\ \Delta s_n(k) \right\}$$

This is used to find $X(k + 1)$ using Kalman filter technique whose computer implementation was described by Fig. 1.

The prediction results obtained by using a hybrid method are tabulated in Tables I and II for 1968-69. Figure 4 shows the comparative results for yearly prediction using (i) only Kalman prediction and (ii) hybrid technique. As seen from these results, system behavior plays an important part in reducing maximum error for yearly prediction (1968 in our case) from 6.37% by Kalman filter to 3.52% by hybrid technique (a reduction by about 44%). While for 1969 an error reduction of about 15% is obtained.

CONCLUSIONS

In this paper a new intermediate-range peak load demand model with state variables $X(k) = \text{col.}\ [\ x(k)\ \ \Delta w(k)\ \ \Delta gd(k)\ \ \Delta s(k)\ ]$ is developed.

The analysis of state Δs suggested that the modification of the system behavior to its normal case would yield better results. Hence a penalty factor algorithm was devised to have such a modification.

A hybrid technique consisting of the combination of Kalman filter state-estimation technique and penalty factor algorithm for penalizing state variable Δs is used for forecasting monthly peak load demands one to three years in advance.

When compared with Kalman estimation, the hybrid technique reduced the yearly forecast error by about 44 percent.

REFERENCES

1. G.B. Berstein and M.J. Cetron, "SEER, a Delphi approach applied to information process," Technological Forecasting, Vol. I, No. 1, June 1969, pp. 33-64.

2. J.P. Martino, "Lognormality of Delphic estimate," Technological Forecasts, Vol. I, No. 4, pp. 355.

3. W.R. Christiaanse, "Short-term load forecasting using general exponential smothering," paper No. 70TP681-PWR, presented at IEEE Summer Power Meeting, July 1970.

4. P.C. Grupte, "Statistical and stochastical techniques for peak power demand forecasting in electric utility systems," PEREC Report 51, Purdue Energy Research and Education Center, Lafayette, Indiana, August 1969.

5. C.R. Williams and J.C. Hubbard, "Application of computer methods to load forecasting," PICA Conference Record, pp. 539, 1965.

6. J. Toyada, M.S. Chen, and Y. Inone, "An application of state estimation to short-term load forecasting," Part I-II, IEEE Trans. on Power Apparatus and Systems, Vol. PAS-80, No. 7, 1970, pp. 1678-82.

7. A.P. Sage, "Optimum systems control," Prentice-Hall, 1968.

8. T.E. Bullock and C.E. Fosha, "A general purpose FORTRAN program for estimation, control and simulation," IEEE Reg. III Conference, Huntsville, Alabama, 1969, pp. 102-07.

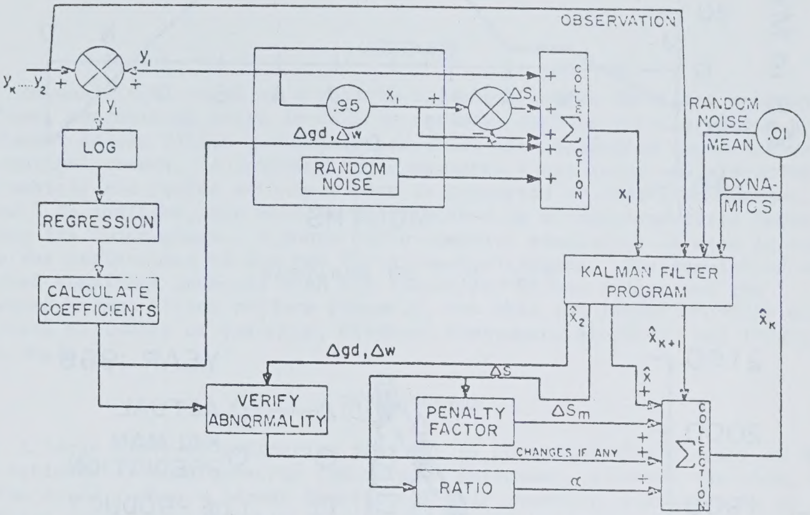

OBSERVATION
y_K ..... y_2
+
y_1
+
y_1
LOG
REGRESSION
CALCULATE COEFFICIENTS
.95
x_1
+
ΔS_1
Δgd, Δw
RANDOM NOISE
+
+
+
+
+
COLLECTION
Σ
x_1
RANDOM NOISE
MEAN
.01
DYNAMICS
KALMAN FILTER PROGRAM
x̂_2
x̂_{K+1}
x̂_K
Δgd, Δw
ΔS
VERIFY ABNORMALITY
PENALTY FACTOR
ΔS_m
CHANGES IF ANY
RATIO
α
x̂
+
+
+
+
COLLECTION
Σ

Fig. 1. Implementation Procedure.

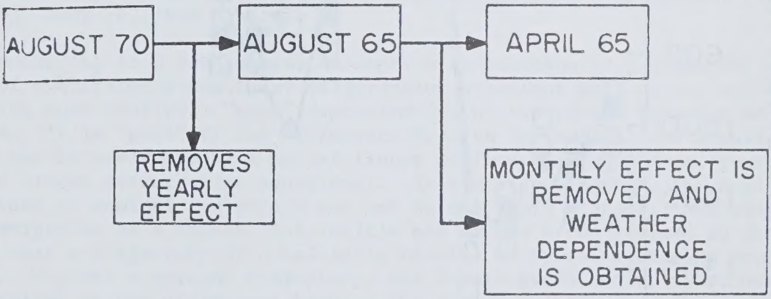

AUGUST 70
AUGUST 65
APRIL 65
REMOVES YEARLY EFFECT
MONTHLY EFFECT IS REMOVED AND WEATHER DEPENDENCE IS OBTAINED

Fig. 2. Data Preprocessing for Time and Weather Independent Data.

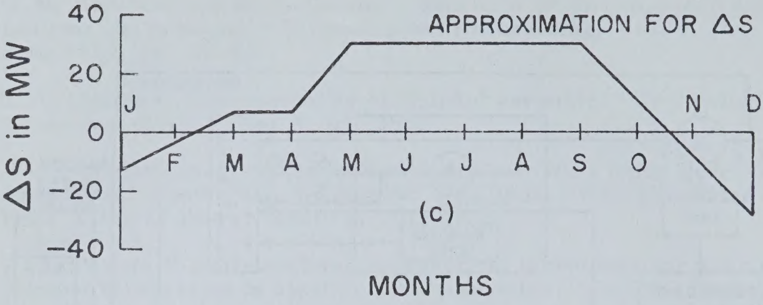


Fig. 3. ΔS Analysis.

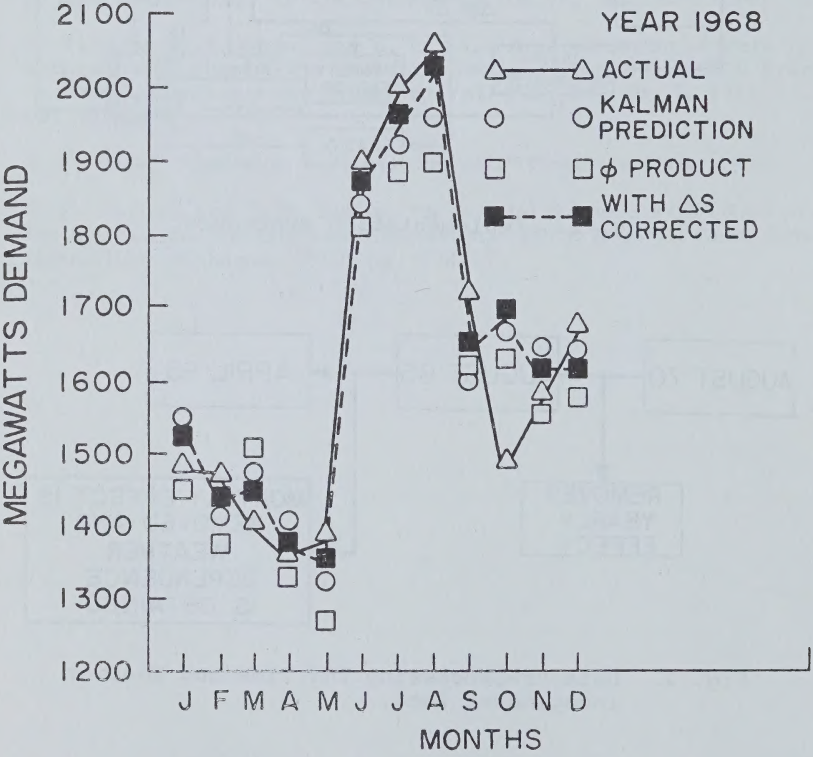


Fig. 4. Comparison of Results.

PERFORMANCE COMPARISON OF SUBOPTIMAL KALMAN FILTERS MODELED FOR A CONTINUOUS NONLINEAR SYSTEM*

HAROLD L. PASTRICK
US Army Missile Command, Redstone Arsenal, Alabama

ABSTRACT

An analytical model of a vehicle launched from an idealized spherical, airless, nonrotating earth is used to estimate various vehicle states by two different Kalman filter mechanizations which are linearized applications of the optimal theory. A tracking radar measures slant range and elevation of the vehicle and yields an output that is corrupted by additive Gaussian noise. In addition, the vehicle is disturbed by a random specific force during its boost phase. A Monte Carlo computer simulation is used to compare the performance of the two filter mechanizations. The results of these limited samplings indicate both the linearized Kalman filter and the extended Kalman filter perform properly, but that the latter provides more accurate estimates of velocity, distance downrange, altitude, and flight path angle.

INTRODUCTION

A large class of estimation problems is concerned with finding an optimal estimate of some quantity (an unknown parameter, a random variable, or a random signal) when a linear function of this quantity, corrupted by an additive noise, is available for generating the estimate. However, the estimation problems most frequently encountered are those in which the unknown quantity is describable by equations that are not linear functions. In these cases the theory of Kalman [2] and Kalman and Bucy [3], developed to indicate optimal state estimates of linear systems, no longer yields optimal estimates. A partial solution, allowing application of their theory in many instances to practical nonlinear systems, was formulated by Kushner [4], Cox [5], Bucy [6], and others.

According to a fundamental assumption by Sorenson [7], a nominal solution of the system's nonlinear differential equations must exist, and this solution must provide a "good" approximation of the actual behavior of the system. It is "good" if the difference between the nominal and actual solutions can be described by a set of linear differential equations (sometimes called linear perturbation equations). In practice, however, the nominal equations or nominal trajectory may not be available *a priori*; for example, the navigation of a terrestrial vehicle may not be predetermined in the sense that a trajectory of a ballistic missile or space booster is predetermined. Without a nominal trajectory, the linear perturbation equations may be obtained as the difference between the actual solution and the current estimate of the nominal trajectory. This process, called relinearization by Bryson and Ho [8], may prove to be even more accurate in estimating the state variable than when linearizing about the nominal.

The purpose of the study described here was to compare the performances of two mechanizations of a pseudo-nonlinear Kalman-Bucy filter: a) a nonlinear system which has been linearized about the nominal, called a linearized Kalman filter [9], and b) one that has been relinearized about the current estimate, called an extended Kalman filter [9] or a relinearized Kalman filter [8].

---

* This paper is a summary of results described more fully in an Army Missile Command publication [1].

MODEL OF THE SYSTEM

The problem is to estimate various parameters for a vehicle launched from an earth assumed to be spherical, airless, and nonrotating. The launch point and trajectory of the vehicle are coplanar with a tracking station located downrange from the launch site. The tracking radar measures slant range and elevation of the vehicle and yields an output which is corrupted by noise. The noisy output is processed to give estimates of the vehicle's states that describe its trajectory, which for this study is assumed simply to be generated by a gravity turn. After launch the rocket is disturbed in the cross track direction by a random specific force, which for realism could be considered a gusting wind disturbance. All the equations for the launch problem can be inferred from the geometry presented in Fig. 1.

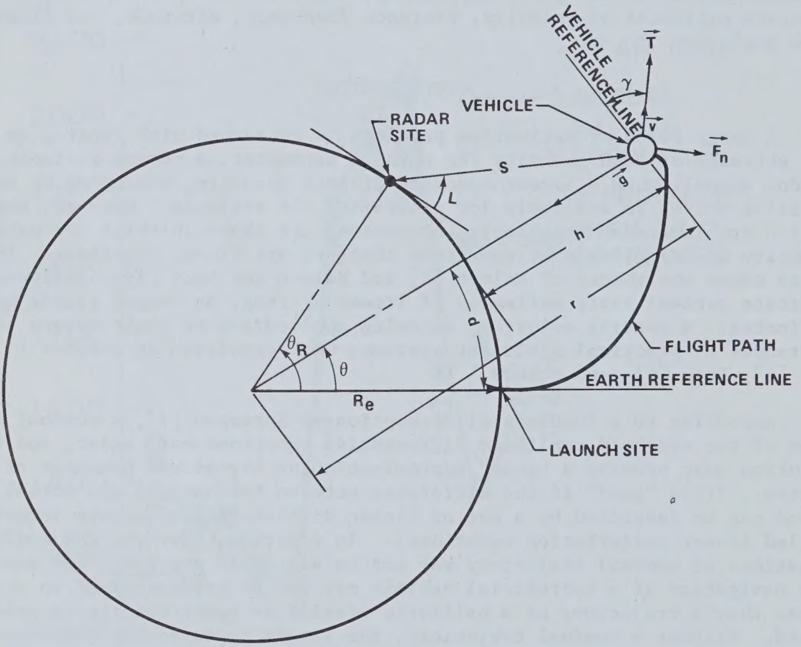


Fig. 1. Geometry of the Problem

The following symbols are applied in Fig. 1:

$R_e \triangleq$ earth's radius

$L \triangleq$ elevation angle between radar site and vehicle

$\theta \triangleq$ reference angle between launch site and vehicle

$\theta_R \triangleq$ reference angle between launch site and radar site

$h \triangleq$ altitude above the earth's surface

$S \triangleq$ slant range distance from radar site to vehicle

$\vec{g} \triangleq$ earth's gravity vector with magnitude at zero altitude

$\vec{v} \triangleq$ velocity of the vehicle with respect to an inertial basis

$\vec{F}_n \triangleq$ disturbing force orthogonal to $\vec{v}$

$\vec{T} \triangleq$ vehicles thrust vector with respect to an inertial basis

$\gamma \triangleq$ flight path angle between velocity vector and a vehicle fixed reference line

$d \triangleq$ distance of the vehicle downrange from the launch site

$r \triangleq h + R_e$.

The following state variable equations defining the vehicle's trajectory are obtainable from the given geometry:

$$\dot{h} = v \sin \gamma \quad , \tag{1}$$

$$\dot{d} = \frac{R_e v \cos \gamma}{h + R_e} \quad , \tag{2}$$

$$\dot{v} = \frac{-k \sin \gamma}{\left(h + R_e\right)^2} + g\left(\frac{T}{w_o}\right) \frac{1}{1 - \left(\frac{T}{w_o}\right) \frac{t}{I_{sp}}} \quad , \tag{3}$$

$$\dot{\gamma} = \frac{d}{R_e} - \frac{k \cos \gamma}{v\left(h + R_e\right)^2} + \frac{F_n}{v} \quad , \tag{4}$$

where

$k \triangleq g R_e^{\,2}$

$t \triangleq$ time of flight measured in seconds from the launch

$I_{sp} \triangleq$ specific impulse of the rocket fuel

$w_o \triangleq$ initial weight of the total vehicle.

The slant range and elevation equations as a function of the states are obtained from simple trigonometric identities. From the law of cosines:

$$S^2 = r^2 + R_e^{\,2} - 2 R_e r \cos\left(\theta_R - \theta\right) \quad , \tag{5}$$

so that

$$S = \sqrt{r^2 + R_e^2 - 2\, R_e\, r \cos\left(\theta_R - \theta\right)} \quad . \tag{6}$$

From the law of sines:

$$\frac{S}{\sin\left(\theta_R - \theta\right)} = \frac{r}{\sin\left(\frac{\pi}{2} + L\right)} = \frac{R_e}{\sin\left(\pi - \left(\frac{\pi}{2} + L\right) - \left(\theta_R - \theta\right)\right)} \quad . \tag{7}$$

If the right equality of Eq. (7) is used, the elevation angle is given by

$$L = \text{arc tan}\left[\frac{\cos\left(\theta_R - \theta\right) - \frac{R_e}{r}}{\sin\left(\theta_R - \theta\right)}\right] \quad . \tag{8}$$

The filter equations are modeled as a continuous nonlinear system. That is, unlike a discrete model, the transition of each state from one increment of time to another is considered to be a smooth continuous process. Furthermore, the observations measured with the radar tracking system are also continuous. Thus the system nonlinear dynamics are expressed by

$$\dot{x} = f[x(t), w(t), t] \quad , \tag{9}$$

where

$x \triangleq$ n-vector of states

$w \triangleq$ m-vector of process noise

$t \triangleq$ time.

The observation equation is also nonlinear and given by

$$z(t) = h[x(t), t] + v(t) \tag{10}$$

where

$z \triangleq$ p-vector of observations

$v \triangleq$ k-vector of measurement noise.

Before expressing the algorithms for the suboptimal filter the following definitions will be useful:

$$F_N(t) \triangleq \left.\frac{\partial f\left[x_N(t), w(t), t\right]}{\partial x_N(t)}\right|_{x_N} \quad , \qquad F_E(t) \triangleq \left.\frac{\partial f\left[\hat{x}(t), w(t), t\right]}{\partial \hat{x}(t)}\right|_{\hat{x}} \quad ,$$

$$G_N(t) \triangleq \left.\frac{\partial f[x_N(t), w(t), t]}{\partial w(t)}\right|_{w_N} , \qquad G_E(t) \triangleq \left.\frac{\partial f[\hat{x}(t), w(t), t]}{\partial w(t)}\right|_{\hat{w}} ,$$

$$H_N(t) \triangleq \left.\frac{\partial h[x_N(t), t]}{\partial x_N(t)}\right|_{x_N} , \qquad H_E(t) \triangleq \left.\frac{\partial h[\hat{x}(t), t]}{\partial \hat{x}(t)}\right|_{\hat{x}} .$$

where

$(\cdot)\big|_{x_N}$ refers to evaluation of the partial derivative at the nominal state.

$(\cdot)\big|_{\hat{x}}$ refers to evaluation of the partial derivative about the estimated state.

Using these definitions the filter equations that describe the state deviations for the two mechanizations follow in Tables I and II. Derivations of these equations can be found in references [2], [3], and [7] through [10].

DISCUSSION

The four trajectory parameters to be estimated are altitude h, downrange distance traveled as measured from the launch site d, velocity v, and flight path angle γ. The launch site and trajectory of the vehicle are in a plane which contains a radar tracking station located 40 nautical miles from the launch site. The simulation is run with the noise w, which is considered a disturbing wind, at a value of zero for the initial 15 seconds. This can be envisioned as a case where the surface winds are negligible for altitudes up to about 2500 feet. The trajectory is planned to achieve an altitude of about 80 miles with a flight path angle suitable for injection into a circular orbit. Thus, at the time of 220 seconds, the ideal case would be to have γ equal to zero.

In the runs where the extended Kalman filter is used, the initial estimated equations are assumed to be the same as the actual equations but without noise. After 10 seconds the estimated equations are generated more accurately by employing the observation modified by the optimal gains, as indicated in Table II. The elevation angle L measured by the tracking radar is not computed for approximately 10 seconds to allow the vehicle, which initially appears below the horizon to the tracker, to become observable. The impact of this delay is to prevent the measurement noise covariance matrix, R, from being computed while it is zero. Computation at zero would cause the computer to reach an exponential overflow, i.e., 1/(R) would become infinite and terminate the simulation prematurely. Several constants, initial conditions, and the 1-σ standard deviations of measurement error had to be chosen. Because this analysis assumes Gaussian statistics, the standard deviation about a zero mean defines the probability distribution function precisely. The values selected are outlined in Tables III and IV.

Finally, the covariance matrices Σ, R, and Q are initialized with the appropriate element given by the square of the standard deviations of Table IV.

Table I. Linearized Kalman Filter Equations

Filter state deviation	$\delta\dot{\hat{x}}(t) = F_N(t)\,\delta\hat{x}(t) + K(t)\left[\delta z(t) - H_N(t)\,\delta\hat{x}(t)\right]$
Suboptimal gain	$K(t) = \Sigma(t)\, H_N^T(t)\, R^{-1}(t)$
Error equation	$\delta\tilde{x}(t) = \delta x(t) - \delta\hat{x}(t)$
Error covariance matrix	$\Sigma(t) = E\left\{\delta\tilde{x}(t),\ \delta\tilde{x}^T(t)\right\}$
Error covariance differential equation	$\dot{\Sigma}(t) = F_N(t)\Sigma(t) + \Sigma(t)F_N^T(t)$ $- \Sigma(t)\, H_N^T(t)\, R^{-1}(t)\, H_N(t)\Sigma(t)$ $+ G_N(t)\, Q(t)\, G_N^T(t)$
Best complete linearized state estimate	$\hat{x}(t) = x_N(t) + \delta\hat{x}(t)$

Table II. Extended Kalman Filter Equations

Filter State equation	$\dot{\hat{x}}(t) = f[\hat{x}(t), t] + K(t)\ [z(t) - h(\hat{x}(t), t)]$
Suboptimal gain	$K(t) = \Sigma(t)\, H_E^T(t)\, R^{-1}(t)$
Error equation	$\tilde{x}(t) = x(t) - \hat{x}(t)$
Error covariance matrix	$\Sigma(t) = E\left\{\tilde{x}(t),\ \tilde{x}^T(t)\right\}$
Error covariance differential equation	$\dot{\Sigma}(t) = F_E(t)\Sigma(t) + \Sigma(t)\, F_E^T(t)$ $- \Sigma(t)\, H_E^T(t)\, R^{-1}(t)\, H_E(t)\Sigma(t)$ $+ G_E(t)\, Q(t)\, G_E^T(t)$

Table III. State Initial Conditions and Constants

States	Constants
$v_o = 100$ ft/sec	$T/w_o = 1.3$
$d_o = 0$ ft	$R_e = 20.89 \times 10^6$ ft
$h_o = 0$ ft	$I_{sp} = 300$ sec
$\gamma_o = 89.66$ deg	$g = 32.17$ ft/sec^2
	$d = 40$ nmi

Table IV. Standard Deviation of Noise

Driving Noise	Measurement Noise
$\sigma_v = 0$ ft/sec	$\sigma_s = 100$ ft
$\sigma_d = 10$ ft	$\sigma_L = 0.01$ rad
$\sigma_h = 100$ ft	
$\sigma_\gamma = 0.001$ deg	

RESULTS AND CONCLUSIONS

The respective performances of the two mechanizations used to estimate the trajectory states are observable in Figs. 2 through 5. Both filter mechanizations performed properly, as evidenced by the fact that the irregular or "noise like" trace is within the $\pm 1\sigma$ curves approximately 67% of the time, which is the percentage suggested by the linearized theory.

An overlay arrangement of the extended filter variance versus the linearized filter variance would show virtually no discernible difference between the two for any of the states estimated. However, the graphic views of the filter errors (Figs. 2-5) indicate the extended Kalman filter mechanization is more accurate than the linearized in estimating velocity v, distance downrange d, and altitude h. The same relationship holds true for the flight path angle γ, but is not as graphically evident.

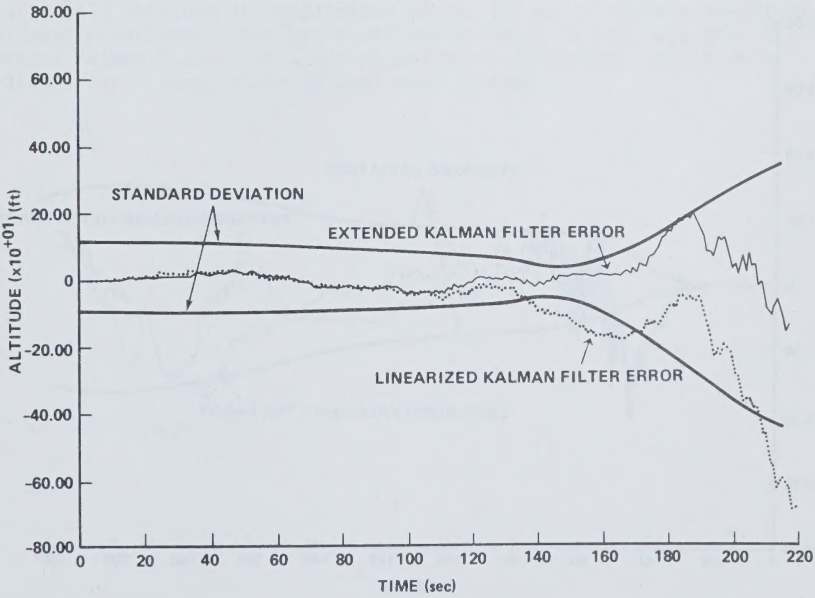


Fig. 2. Comparison of Filter Performance (Altitude)

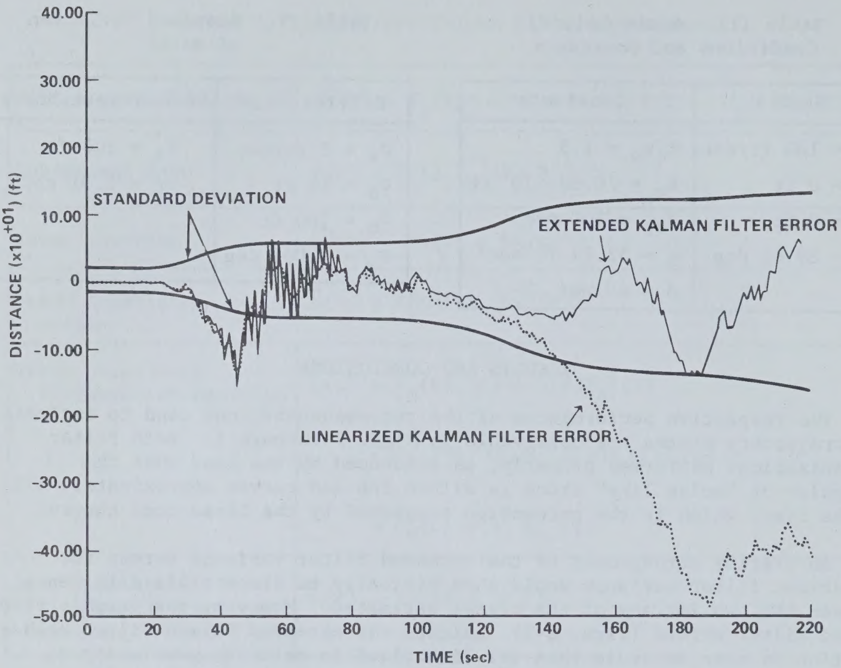


Fig. 3. Comparison of Filter Performance (Distance)

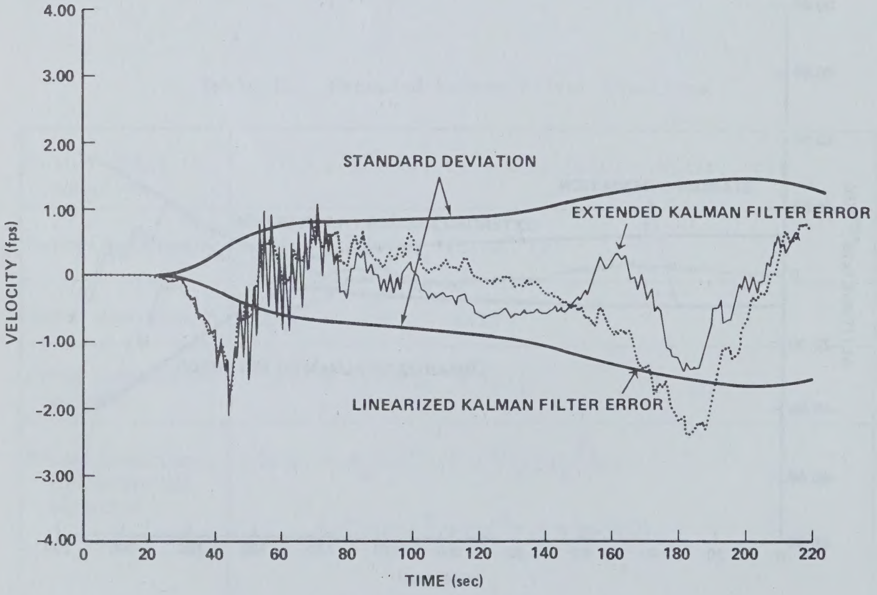


Fig. 4. Comparison of Filter Performance (Velocity)

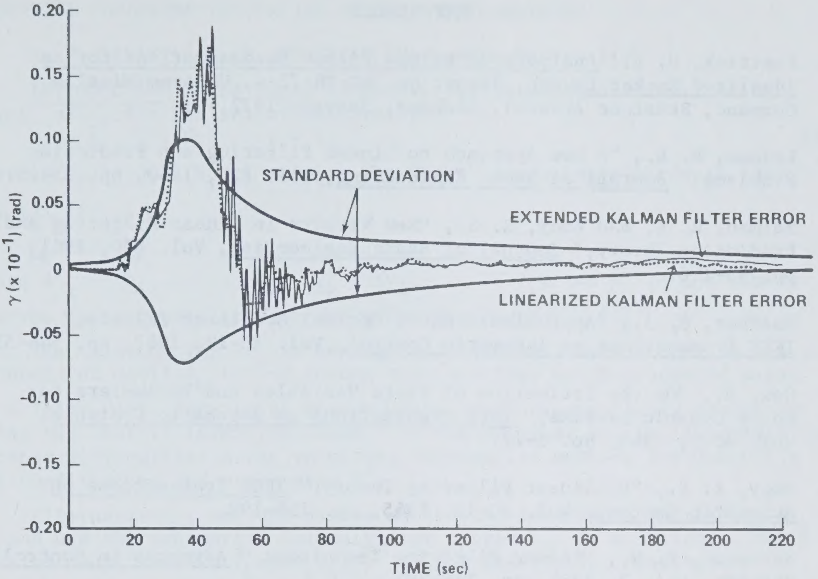


Fig. 5. Comparison of Filter Performance (Angle of Attack)

The finding that the extended Kalman filter performs with less error than the linearized Kalman filter appears to be in agreement with other results [10] obtained in application of the filter to systems described by nonlinear equations. Thus it is not unreasonable to conclude that the extended Kalman filter, as a suboptimal state estimator, is a viable candidate for a large class of nonlinear systems.

REFERENCES

1. Pastrick, H. L., Analysis of Kalman Filter Mechanizations for an Idealized Rocket Launch, Report No. RG-TR-72-4, US Army Missile Command, Redstone Arsenal, Alabama, January 1972.

2. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, Vol. 82D, 1960, pp. 35-45.

3. Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," Journal of Basic Engineering, Vol. 83D, 1961, pp. 95-108.

4. Kushner, H. J., "Approximations to Optimal Nonlinear Filters," IEEE Transactions on Automatic Control, Vol. AC-12, 1967, pp. 546-556.

5. Cox, H., "On the Estimation of State Variables and Parameters for Noisy Dynamic Systems," IEEE Transactions on Automatic Controls, Vol. AC-9, 1964, pp. 5-12.

6. Bucy, R. S., "Nonlinear Filtering Theory," IEEE Transactions on Automatic Control, Vol. AC-10, 1965, pp. 198-199.

7. Sorenson, H. W., "Kalman Filtering Techniques," Advances in Control Systems, Vol. 3, 1966, pp. 219-292.

8. Bryson, A. E., Jr. and Ho, Y. C., Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

9. Sage, A. P. and Melsa, J. L., Estimation Theory with Applications to Communications and Control, McGraw-Hill Book Company, New York, 1971.

10. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

SEQUENTIAL PARAMETER ESTIMATION USING PSUEDO-INVERSE*

R. P. NAYAK** & E. C. FOUDRIAT, Marquette University

INTRODUCTION This paper considers the least-squares estimation (LSE) of $\underline{X}$, for the set of observations

$$y_i = \underline{h}'_i \, \underline{X} + N_i \qquad i = 1, \ldots \tag{1}$$

where $E(N_i) = 0$, $E[N_i N_j] = \delta_{ij}$. The "best" approximate estimate (BAE) for (1) is

$$\hat{\underline{X}}_k = A_k^{+} \underline{Y}_k \tag{2}$$

where $A'_k = (\underline{h}_1, \ldots, \underline{h}_k)$, $\underline{Y}_k = (y_1, \ldots, y_k)$ and A^+ is the psuedo-inverse (alt., generalized or Moore-Penrose Inverse). [1]

The solution for A^+_k is generally difficult. In most LSE, a set is accumulated until A_k is of column rank and then batch-processed using $A^+ = (A'A)^{-1}A'$. Additional measurements can be sequentially processed using the matrix inversion lemma. If the original set is ill-conditioned further difficulties arise requiring alternative methods for detecting ill-conditioning and processing A^+.

Alternatively, one can assume an initial covariance matrix $E[\underline{X}_o \underline{X}'_o] = P_o$ and use the sequential Kalman filter approach. If observable (well-conditioned), the $\lim\limits_{k\to\infty} P_k \to (A'_k A_k)^+$. If not observable, convergence is not guaranteed, so the BAE is not assured.

This problem could be eliminated by a LSE that:

(1) is a sequential BAE at all steps (no switch from batch to sequential or aprioni assumption of P_o)
(2) easily recognizes ill-conditioning and implements an on-line decision.

SEQUENTIAL PSUEDO-INVERSE ESTIMATOR The process is based upon a theorem by Cline [2] for psuedo-inverse of partitioned matrices

$$\begin{aligned} P_{k+1} &= \left[(A_k' \,\vdots\, \underline{h}_{k+1}) \begin{pmatrix} A_k \\ \hdashline \underline{h}'_{k+1} \end{pmatrix}\right]^+ \\ &= (I_n - \underline{D}_k'^{+} \underline{h}_{k+1}) \, [P_k \underline{h}_{k+1} (1 - \underline{D}_k^{+} \underline{D}_k) \, K \underline{h}'_{k+1} \, P_k] \cdot \\ &\qquad (I - \underline{h}_{k+1} \underline{D}^{+}) + \underline{D}_k^{+'} \underline{D}_k^{+} \end{aligned} \tag{3}$$

$$\underline{D}_k = (I_n - A_k^{+} A_k) \underline{h}_{k+1} = (I_n - P_k A_k' A_k) \underline{h}_{k+1}$$

$$K = [1 + (1 - \underline{D}_k^{+} D_k) \underline{h}_{k+1} \; P_k \underline{h}_{k+1} \; (1 - \underline{D}_k^{+} \underline{D}_k)]$$

Based upon the property that $(I_n - \underline{A}_k^{+} A_k)$ is in the null space of A, $\underline{D}_k = 0$ iff rank (A_{k+1}) = rank (A_k), i.e., $\underline{h}_{k+1}$ does not increase the dimension of A_{k+1}, (3) can be simplified. Three cases must be considered:

(1) All parameters are observable, rank $(A_k) = n$ (Column rank)
(2) Some parameters are unobservable, rank $(A_k) < n$ and rank (A_{k+1}) = rank (A_k)

---

*NASA, Langley Research Center Grant, NGR 50-001-009

**Dr. Nayak is with Rexham Corporation, Rockford, IL

(3) Some parameters are unobservable but rank $(A_{k+1}) >$ rank (A_k)

Using (3) and the condition on $\underline{D}_k$, the sequential BAE is summarized in Table 1. Case 1 and 2 are identical to the Kalman filter, as expected by [3] and [4]. For Case 3, the estimator $\underline{X}_{k+1}$ is identical to 1 & 2, but P_{k+1} differs.

$\underline{D}_k$ is critical in processing cases 2 & 3. Hence the decision criteria $|D_k| < \varepsilon$, case 2; $|\underline{D}_k| > \varepsilon$, case 3; ε is selected based on system errors. For calculating $\underline{D}_k$ at each step in cases 2 & 3, $A'_k A_k = A'_{k-1} A_{k-1} + \underline{h}_k \underline{h}'_k$ is retained. Once rank $(A_k) = n$, $\underline{D}_k \equiv 0$ reverting to Case 1.

The system fulfills the stated objectives of a completely sequential BAE with no apriori assumptions and an on-line, ill-conditioned decision criteria.

References

[1] Penrose, R., "Generalized Inverse for Matrices", Proc. CAMB. Phil. Soc. 51, (1955), pp. 406-13.

[2] Cline, R. H., "Representation for the Generalized Inverse of Sums of Matrices", J. SIAM Num. Ana. 12, 1, (1965), pp. 99-114.

[3] Lewis, T., Odell, P., Estimation in Linear Models, Prentice-Hall, 1971 Ch. 1 & 3.

[4] Aoki, M. Optimization of Stochastic Systems, Acad. Press. 1967, Ch 5.

Table 1 Summary of Sequential Psuedo-Inverse Estimation

Estimate of	Case #1 & #2	Case #3
	Observable rank $A_k = n$ Unobservable rank $(A_k) = $ rank $(A_{k+1}) < n$	Unobservable rank $(A_k) = r$, rank $(A_{k+1}) = r+1$
$\hat{\underline{X}}_{k+1} =$	$\hat{X}_k + P_{k+1}\underline{h}_{k+1}(y_{k+1} - \underline{h}'_{k+1}\hat{X}_k)$	$\hat{X}_k + P_{k+1}h_{k+1}(y_{k+1} - h_{k+1}\hat{X}_k)$
$P_{k+1} =$	$P_k - P_k\underline{h}_{k+1} K \underline{h}'_{k+1} P_k$ where $\underline{D}_k = 0$ $K = (1 + \underline{h}'_{k+1} P_k \underline{h}_{k+1})^{-1}$ and $P_{k+1}\underline{h}_{k+1} = P_k \underline{h}_{k+1} K$	$(I - \underline{D}^{+'}_k \underline{h}'_{k+1}) P_k (I - \underline{h}_{k+1}\underline{D}^{+}) + D^{+'}_k D^{+}_k$ where $\underline{D}_k = (I - P_k A'_k A_k)\underline{h}_{k+1} \neq 0$ and $P_{k+1}\underline{h}_{k+1} = \underline{D}^{+'}_k$

ON THE ESTIMATION AND DETECTION FOR OPTICAL TURBULENT CHANNELS*

ROBERT O. HARGER and KOJI KURIMOTO
Department of Electrical Engineering
University of Maryland, College Park, Maryland 20742

This paper considers the problem of filtering and detecting signals received through a turbulent channel at optical frequencies. Adopting the semi-classical method of quantum-mechanics, application is made to the direct detection of a single path optical communication channel where the received field is assumed to be the coherent signal field degraded by the atmospheric turbulence with a signal-independent, additive zero mean Gaussian background radiation.

Previous investigations, which have been made assuming that the observed data is a doubly stochastic Poisson counting process with the rate parameter of which is proportional to the sum of the intensities by the transmitted laser beam modulated by an atmospheric turbulence and by a small background radiation (Snyder [1], Clark [2], Evans [3]), are now extended to include the effects of the detector' s response and thermal noise.

Assuming that the observed data is the superposition of a doubly stochastic shot noise process, due to the interaction of the received field with the photodetector, and a white Gaussian thermal noise, the problem is formulated using the state space representation for the observed data and the shot noise process which is assumed to be generated by a first order linear system driven by a doubly stochastic Poisson process with the rate parameter modulated by a vector-valued general Markov process modelling the state of the atmospheric turbulence.

The basic equation which describes the time evolution of the posteriori statistics of the shot noise process as well as those of the state of turbulence are derived using a stochastic differential form of the conditional joint characteristic function. Directly then the minimum mean square error estimator equations for the shot noise process and the state of turbulence are obtained. Using a certain linearization method, or "quasi-linear" approximation proposed by Snyder [1], the "quasi-optimum" receiver in the form of the "general likelihood ratio" (Kailath [4]) is specified in its most general form.

In order to evaluate the quality of a specified receiver, a new method of calculating the probability of error performance for the general likelihood ratio is derived in terms of the solution of a Fokker-Planck equation. This method is applied to calculate the probability of error of a proposed sub-optimum receiver for a frozen atmosphere, spatially invariant, log-normal fading channel, which utilizes the current estimate of the shot noise process. Using the approximation method proposed by

---

*Research supported by NASA Grant NGR 21-002-237 and NSF Grant GK-14920.

Stratonovich [5] to approximate an actual process by a stochastically equivalent Markov process, the approximate probability of error for the sub-optimum receiver is calculated.

REFERENCES

[1] D. L. Snyder, "Filtering and Detection for Doubly Stochastic Poisson Processes", IEEE Trans. on Info. Theo., Vol. IT-18, pp. 91, 1972.

[2] J. R. Clark, "Estimation for Poisson Processes", MIT Quarterly Progress Report, No. 100, pp. 146, 1971.

[3] E. Evans, "Likelihood Ratio and Mutual Information for Poisson Processes with Stochastic Intensity Functions", MIT Quarterly Progress Report, No. 98, pp. 103, 1971.

[4] T. Kailath, "A General Likelihood Ratio Formula for Random Signals in Gaussian Noise", IEEE Trans. on Info. Theo., Vol. IT-15, pp. 350, 1969.

[5] R. L. Stratonovich, "Topics in the Theory of Random Noise", Gordon and Breach, Vol. 1, New York, 1963.

ON THE ESTIMATION PROBLEM AS VIEWED FROM INFORMATION THEORY

P. KALATA
Ill. Institute of Technology
Chicago, Illinois

R. PRIEMER
University of Illinois
Chicago, Illinois

INTRODUCTION

Historically, information theory has been used in the analysis of communication systems. Shannon [1] formulated a theory which emphasizes probability theory, establishes the existence of codes, and has been extended to allow for the development of practical codes.

This paper is concerned with applying information theory to the estimation problem. Some advantages sought with this approach are: (1) provide insight into the form of good estimators, (2) establish bounds on estimator performance; (3) unify the areas of communication and estimation theory. For example, recently Zakai and Ziv [2] have applied informational concepts to yield bounds on the performance of a class of estimators.

MEASUREMENT OF INFORMATION

Mutual information between the random variables x and z is defined as the logarithmic ratio of an a posteriori and an a priori density function,

$$i(x;z) = \log(p(x/z)/p(x)),$$

and measures [3] the extent to which the occurrence of a value of z is related to the value of x. If z uniquely specifies x, then mutual information reduces to self information,

$$h(x) = \log(1/p(x)).$$

Self information may be interpreted either as the a priori uncertainty of x or as the amount of information required to resolve the uncertainty. Contrary to what the term may imply, self information is a measure of what is not known about x.

Mutual and self information are both random variables. The average self information H(x), which is termed the entropy of the ensemble, and the average mutual information I(x;z) are most significant. For example, a small value of entropy indicates that the probability density function is concentrated at some value of x.

ON THE ESTIMATION PROBLEM

In the Bayesian approach to the estimation problem, a noisy observation z of a state x is given and an estimate $\hat{x}(z)$ is sought such that some average cost function $L(\tilde{x} = x - \hat{x})$ is minimized. If the cost function is the error self information, then Bayes risk is the error entropy $H(\tilde{x})$. In the limiting process, if $p(\tilde{x}) \to \delta(\tilde{x})$, then $H(\tilde{x}) \to -\infty$ for a perfect unbiased estimator. Under Gaussian statistics, the cost function $\log(1/p(\tilde{x}))$ is convex U(cup). Therefore, using the results of Sherman [4], the minimum error entropy and the minimum mean square error estimators are identical.

With error entropy as the performance criterion, Weidemann [5] has shown that: (1) the equivocation, which is the conditional entropy H(x/z) and is independent of the estimator, is the lower bound for $H(\tilde{x})$, (2) min. $H(\tilde{x})$ is equivalent to min. $I(\tilde{x};z)$, and (3) the lower bound is achieved when $\tilde{x}$ and z are statistically independent.

An estimator may be designed by postulating its form, obtaining $H(\tilde{x})$, and minimizing $H(\tilde{x})$ to determine the estimator parameters. The performance of the postulated estimator can be evaluated by comparing $H(\tilde{x})$ and H(x/z). To yield minimal $I(\tilde{x};z)$, a necessary condition for the estimator parameters is

$$0 = \int_{-\infty}^{\infty} \left\{ \ln\left(\frac{e\, p(\tilde{x}/z)}{p(\tilde{x})}\right) \nabla_p\, p(\tilde{x}/z) - \frac{p(\tilde{x}/z)}{p(\tilde{x})} \nabla_p\, p(\tilde{x}) \right\} d\tilde{x},$$

where ∇_p is the gradient operator with respect to the estimator parameter set.

EXAMPLE

Consider the discrete system and measurement procedure

$$x(k+1) = \Phi(k+1, k)x(k) + \Gamma(k+1, k)u(k)$$

$$z(k) = D(k)x(k) + w(k),$$

along with the usual assumptions on $u(k)$, $w(k)$, and $x(0)$. Under Gaussian statistics, it can easily be shown that a postulated linear estimator which minimizes the error entropy $H(\tilde{x}(k/k))$ with respect to the estimator parameters is also the conditional mean estimator, and that $H(\tilde{x}(k/k)) = H(x(k)/\underline{z}(k))$, where $\underline{z}(k) = \{z(k), z(k-1), \cdots, z(0)\}$.

To obtain a recursive estimator, note that min. $H(\tilde{x}(k+1/k+1))$ is equivalent to min. $I(\tilde{x}(k+1/k+1); \underline{z}(k+1))$, and make use of the identity

$$I(A;B) = H(A) - H(A/B), \tag{1}$$

and therefore

$$I(\tilde{x}(k+1/k+1); \underline{z}(k+1)) = 0.$$

With the use of the identity

$$I(A;BC) = I(A;B) + I(A;C/B),$$

and since average mutual information is always non-negative, the requirement that $H(\tilde{x}(k+1/k+1))$ take on its minimum possible value, yields

$$I(\tilde{x}(k+1/k+1); \underline{z}(k)) = 0 \tag{2}$$

$$I(\tilde{x}(k+1/k+1); z(k+1)/\underline{z}(k)) = 0. \tag{3}$$

With the further use of eq. (1), eq. (2) yields the structure for the filter and a recursive relationship for the filter parameters and eq. (3) determines parameter values. The resulting recursive filter is identical to the well known Kalman-Bucy filter.

FUTURE WORK

For a class of estimation problems it can be shown that the minimum error entropy estimate is the conditional mean. Also of interest is the equivocation for smoothing estimators. These results will be reported elsewhere.

CONCLUSION

The concept of uncertainty has an intuitive appeal as an estimation performance criterion. In addition, of significance is that this approach yields a lower bound for estimation performance. However, a disadvantage is that the calculation of the system's equivocation for nonlinear systems or non-Gaussian disturbances compares in difficulty to the calculation of the conditional mean.

REFERENCES

1. Shannon, C. E. and Weaver, W., 1949, The Mathematical Theory of Communication, U. of Ill. Press, Urbana, Ill.
2. Zakai, M. and Ziv, J., 1972, "Lower and Upper Bounds on the Optimal Filtering of Certain Diffusion Processes", IEEE Trans. Inf. Thy., IT-18, pp. 325.
3. Gallager, S., 1968, Information Theory and Reliable Communication, Wiley, New York.
4. Sherman, S., 1958, "Non-Mean-Square Error Criteria", IEEE Trans. Inf. Thy., IT-4, pp. 125.
5. Weidemann, H. L., 1969, "Entropy Analysis of Feedback Control Systems", Advances in Control Systems, C. T. Leondes (ed.), pp. 225.

A MULTIPLE FAULT DETECTION ALGORITHM FOR COMBINATIONAL LOGIC CIRCUITS*

S. S. YAU AND S. C. YANG
Departments of Computer Sciences and Electrical Engineering
and Biomedical Engineering Center
Northwestern University
Evanston, Illinois 60201

ABSTRACT

A general algorithm for detecting multiple stuck-at faults in combinational logic circuits is presented. This algorithm has about the same complexity as the simplest existing general multiple fault detection algorithm, but the test sets generated by this algorithm always have fewer tests for redundant logic circuits, and have no more tests for irredundant logic circuits than those generated by the existing algorithm.

I. INTRODUCTION

One of the most important problems in fault-tolerant computing is to detect whether there exist faults in the various components of the computing system. Fault detection should be completed as quickly as possible, especially in a real-time environment, so that the system, after detecting the existence of faults in some components, can still provide acceptable services by proper reconfiguration. Consequently, fault detection test sequences should be as short as possible, and they are applied on an interrupt basis. Combinational logic circuits are vital parts of the computing systems, and hence fault detection of combinational logic circuits has attracted much attention [1] - [8]. It is noted that for a combinational logic circuit we only need to consider fault detection test sets instead of test sequences because the order of the tests in a test set to be applied to a combinational logic circuit is immaterial.

So far, most of the fault detection techniques are under the single stuck-at fault assumption and only a few [3] - [8] have dealt with multiple stuck-at faults. Under the single stuck-at fault assumption, the problem of generating a fault-detection test set is relatively simple. Since a physical fault in the logic circuits usually cannot be represented by a single stuck-at fault, but it may often be represented by a multiple stuck-at fault [9], the fault detection test sets generated under the single stuck-at fault assumption are frequently not sufficient in practice. Furthermore, a detectable single stuck-at fault may not be detected by a test set generated under the single stuck-at fault assumption when its occurence is preceded by the occurrence of an undetectable fault [2],[3]. Therefore, it is necessary to consider multiple stuck-at fault detection. However, most existing stuck-at multiple fault-detection techniques are heuristic in nature [3], [8] or applicable only to special types of logic circuits [5], [7] . Some general multiple stuck-at fault-detection algorithms are also available, but they are too complicated in computation and generate test sets containing too many tests [4] - [6].

In this paper, we shall present a general algorithm for detecting multiple stuck-at faults. This algorithm has about the same complexity as that of the simplest existing algorithm [5], but the test sets generated by it always have fewer tests for redundant logic circuits, and have no more tests for irredundant logic circuits than those generated by [5].

II. PRELIMINARIES

Let $F = F(x_1,\ldots,x_n)$ be the output function of a combinational logic circuit. Let α be an arbitrary stuck-at fault in the logic circuit, which changes the output function F to F*, where $F^* = F^*(x_1,\dots,x_n,\alpha)$. α is said to be detectable if $F^* \neq F$ and undetectable if $F^* = F$. The function F* is called a fault function of α. Two faults in a logic circuit are said to be functionally equivalent if and only if their fault functions are equal. Let us consider a logic function as a set of minterms. For logic functions $F_1,\dots,F_h$ of variables $x_1,\dots,x_n$, a minimum cover of $F_1,\dots,F_h$ is defined as a minimal set of minterms, which has at least one minterm in each of $F_1,\dots,F_h$. An input pattern to a logic circuit producing different outputs when the circuit has the fault α or is fault-free is called an input test pattern of α. A test of α is an input test pattern of α with the corresponding output of the fault-free circuit. A test set for detecting all the faults under consideration is a set which contains at least one test for each of the faults under consideration.

---

*The work reported here was partially supported by Public Health Service Grant 5 P01 GM 15418.

Let a_k be either 0 or 1, and $x_k^0 = \overline{x}_k$ and $x_k^1 = x_k$. We define $(a_1, a_2, \dots, a_n)$ as the input pattern represented by the minterm $x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}$. For a fault-free logic circuit with the output function F, application of an input pattern represented by any minterm of F or $\overline{F}$ yields the output 1 or 0 respectively.

III. AN ALGORITHM FOR GENERATING MULTIPLE FAULT-DETECTION TEST SETS

Let $\mathcal{J}^* = \{F_1^*,\dots,F_r^*\}$ be the set of all distinct fault functions of all the detectable faults under consideration. It is well known [1] that, if $F_i^* \in \mathcal{J}^*$ and F is the output function of a combinational circuit without any fault , then $F_i^* \oplus F = F_i^*\cdot\overline{F} + \overline{F_i^*}\cdot F$ represents all possible input test patterns to detect all the faults with the fault function F_i^*. Hence, an optimal fault-detection test set can be obtained by finding a minimum cover of $F_i^* \oplus F$, $i = 1,\dots,r$, where an optimal fault-detection test set is defined as a fault-detection test set having a smallest number of tests. It is noted that not all the fault functions in $\mathcal{J}^*$ are required in generating the minimum cover of $F_i^* \oplus F$, $i = 1,\dots,r$. For example, if $F_i^* \oplus F \supseteq F_j^* \oplus F$, where $F_i^*, F_j^* \in \mathcal{J}^*$, then F_i^* need not be considered. Therefore, it is desired to find a minimum set of fault functions, $M^* \subseteq \mathcal{J}^*$, which is sufficient for the generation of fault-detection test sets. Unfortunately, there is no good method to find such a set. However, we could find a set of functions $\mathcal{R}^* \supseteq M^*$ for generating the fault-detection test set such that the number of elements in $\mathcal{R}^*$ is much smaller than that in $\mathcal{J}^*$. To do this, let us first classify $\mathcal{J}^*$ into the following two classes:

$$\mathcal{G}_1^* = \{G^*_{1i} \mid G^*_{1i} \subset F\}$$

$$\mathcal{G}_2^* = \{G^*_{2i} \mid G^*_{2i} \cdot \overline{F} \neq 0\}$$

In particular, $F^* = 1$ belongs to $\mathcal{G}_2^*$ and $F^* = 0$ belongs to $\mathcal{G}_1^*$. It is obvious that the above two classes are disjoint and their union is the set $\mathcal{J}^*$. In generating fault detection test sets, $F^* = 0$ (or 1) can be excluded since it can be detected by the input test pattern represented by any minterm in F (or $\overline{F}$).

Let us define $\mathcal{H}_1^* = \{H^*_{1i}\}$ as the representative function set of $\mathcal{G}_1^*$ such that for every $G^*_{1i} \in \mathcal{G}_1^*$, there exists an $H^*_{1j} \in \mathcal{H}_1^*$ so that $\overline{H^*_{1j}}\cdot F \subseteq \overline{G^*_{1i}} \cdot F$; and $\mathcal{H}_2^* = \{H^*_{2i}\}$ as the representative function set of $\mathcal{G}_2^*$ such that for every $G^*_{2i} \in \mathcal{G}_2^*$, there exists an $H^*_{2j} \in \mathcal{H}_2^*$ so that $H^*_{2j}\cdot\overline{F} \subseteq G^*_{2i}\cdot\overline{F}$. Then, it can be shown that the union of the minimum cover of $(\overline{H^*_{1i}}\cdot F)$'s and the minimum cover of $(H^*_{2i}\cdot\overline{F})$'s represents the near minimum input pattern test set. If $\mathcal{H}_1^*$ and $\mathcal{H}_2^*$ are both contained in $\mathcal{T}^*$, the number of input test patterns in the test set is more close to the minimum. Otherwise, the resulting test set will have more redundant tests. In summary, the algorithm for generating multiple fault-detection test set can be stated as follows:

A-1) Find the representative function sets $\mathcal{H}_1^*$ and $\mathcal{H}_2^*$ of $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ for the given logic circuit. Let $\mathcal{H}_1^* = \{H^*_{1i}\}$, $i = 1,\ldots,p$, and $\mathcal{H}_2^* = \{H^*_{2i}\}$, $i = 1,\ldots,q$.

A-2) Find the minimum cover of $\overline{H^*_{1i}}\cdot F$, $i = 1,\ldots,p$, and the minimum cover of $H^*_{2i}\cdot\overline{F}$, $i = 1,\ldots,q$.

A-3) Form a test set T by first taking the union of minimum covers obtained in Step A-2) and then substituting the minterms in that union with the corresponding tests.

In the remainder of this paper, we shall present a method to generate the representative function sets $\mathcal{H}_1^*$ and $\mathcal{H}_2^*$ for a given logic circuit.

IV. GENERATION OF REPRESENTATIVE SETS $\mathcal{H}_1^*$ and $\mathcal{H}_2^*$

It has been shown [5] that all multiple stuck-at faults in a logic circuit are equivalent to the stuck-at faults on the check lines of the circuit, where a check line is either a fan-out branch or a primary input line which is not a fan-out line. Consequently, we only need to consider the stuck-at faults on the check lines. Our approach to generate $\mathcal{H}_1^*$ and $\mathcal{H}_2^*$ is based on the manipulation of the terms of a function F_c called a check function. The check function F_c is a function of the check line symbols and input variables, which can be constructed as follows:

B-1) For the given logic circuit, label all the check lines $c_1, c_2, \ldots$ and the other lines $d_1, d_2, \ldots$

B-2) Consider the labels of the lines as the logic symbols on the lines. Let F_c equal to the primary output line symbol which is in turn expressed in terms of the input line symbols of the output gate. Now, repeat this process for each line symbol until all the primary input lines or the fan-out branches of the primary input lines are reached. At each iteration of expressing a line symbol in terms of other line symbols through a gate, three rules are followed: (i) If the line is a check line, say c_j, then c_j is replaced by the logic product of c_j and the logic expression of c_j in terms of the input line symbols of the gate. If the line is not a check line, say d_i, then d_i is simply replaced by the logic expression of d_i in terms of the input line symbols of the gate. (ii) The complement operation should only apply to individual line symbols. For instance, with the input lines c_j and c_{j+1}, the output line symbol of a NOR gate should be written as $\overline{c}_j\cdot\overline{c}_{j+1}$ instead of $\overline{c_j + c_{j+1}}$, and the output line symbol of a NAND gate should be written as $\overline{c}_j + \overline{c}_{j+1}$ instead of $\overline{c_j c_{j+1}}$. (iii) Skip each fan-out line by considering each of its fan-out branches as the fan-out line.

B-3) Replace the symbol of each primary input line (or each fan-out branch of a primary input line) by the product of the symbol and the corresponding primary input variable. In this process, the term with the product of an input variable and its complement should be retained, that is, the rule $x_i\overline{x}_i = 0$ is not used in this algorithm. The check function F_c so constructed has the form

$$F_c = \Sigma_k X_k Y_k,$$

where each X_k is a product of some primary input variables (or their complements) and Y_k is a product of the check line symbols (or their complements).

It is obvious from the above algorithm that the output function F of a logic circuit without any fault can be expressed as $F = \sum_k X_k$.

Now, let us examine the effect of a multiple stuck-at fault α on a term X_kY_k of F_c, denoted as $(X_k)\alpha$. To do this, the following notations will be used. Let b_j be either 0 or 1 and $c_j^0 = \bar{c}_j$ and $c_j^1 = c_j$. A check line symbol can thus be expressed as $c_j^{b_j}$ and a single check line stuck-at-b_j fault can be expressed as $c_j^{(b_j)}$. A multiple stuck-at fault can be denoted as $\alpha = (c_{j1}^{(b_{j1})}, c_{j2}^{(b_{j2})}, \ldots, c_{jr}^{(b_{jr})})$. For each check line c_j with a stuck-at fault in the multiple stuck-at fault α, X_kY_k can be modified as follows: (i) When c_j is a primary input line or a fan-out branch fanned out from a primary input line, if $c_j^{b_j} \in Y_k$ and line c_j has a stuck-at-b_j fault, then substitute $c_j^{b_j}$ and the input variable (or its complement) applied to line $c_j^{b_j}$ by 1 in X_kY_k; if $c_j^{b_j} \in Y_k$ and line c_j has a stuck-at-$\bar{b}_j$ fault, then set $X_kY_k = 0$. (ii) When c_j is a fan-out branch from an output line, if if $c_j^{b_j} \in Y_k$ and line c_j has a stuck at-b_j fault, then substitute $c_j^{b_j}$ and $c_r^{b_r}$'s and the input variables or their complements on which the value of $c_j^{b_j}$ depends by 1 in X_kY_k: if $c_j^{b_j} \in Y_k$ and line c_j has a stuck-at-$\bar{b}_j$ fault, then set $X_kY_k = 0$. After repeating this modification, fan-out branches first, for each check line c_j with the stuck-at fault in α, we obtain $(X_k)\alpha$ by substituting all the remaining check line symbols in Y_k by 1. Considering X_k and $(X_k)\alpha$ as sets of minterms, there are four possibilities for $(X_k)\alpha$:

$$(1) \quad (X_k)\alpha = 0,$$
$$(2) \quad (X_k)\alpha \cdot \bar{F} \neq 0,$$
$$(3) \quad (X_k)\alpha = X_k,$$
$$(4) \quad X_k \subset (X_k)\alpha \subseteq F.$$

In ordinary logic circuits the chance that $X_k \subset (X_k)\alpha \subseteq F$ is very small. Let us define a Type 1 logic circuit as a logic circuit for which none of the possible multiple stuck-at faults will cause $X_k \subset (X_k)\alpha \subseteq F$; and all the other logic circuits are called Type 2 logic circuits . It can be shown that a non-redundant logic circuit is a Type 1 circuit. To simplify the presentation, the following algorithm to find $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ for Type 1 logic circuits only is presented. In this algorithm, a J-component fault is defined as a fault with J single check line component stuck-at faults. We shall first construct the Jth tables, J = 1,2,..., as follows

(i) The Jth tables are constructed in the order J = 1,2,....

(ii) Each term X_k in F_c forms a column of the Jth table.

(iii) A J-component fault α_{ji} forms a row if all the (J-1)-component faults corresponding to all combinations of (J-1) components of α_{ji} are left in the (J-1)th table. When J = 1, the single check line fault $c_j^{(\bar{b}_j)}$ corresponding to each check line symbol $c_j^{b_j}$ in F_c forms a row of the 1st table.

(iv) If $(X_k)\alpha_{ji} = 0$, enter a 1 into the entry of X_k and α_{ji}; if $(X_k)\alpha_{ji} = X_k$ enter a 0 into the entry of X_k and α_{ji}; and if $(X_k)\alpha_{ji} \cdot \overline{F} \neq 0$, enter a 1* into the entry of X_k and α_{ji}.

(v) If a row does not have any 1*'s, calculate the weight for it by counting the number of 1's in the row.

Also, the lists Q_J for a term $X_i Y_i$ in F_c are constructed as follows:

(i) The lists are constructed in the order $J = 1,2,\ldots$.

(ii) A J-component fault α_{jm} is entered into Q_J if all the (J-1)-component faults corresponding to all the combinations of (J-1) components of α_{jm} are left in Q_{J-1}. When $J = 1$, all the single check line faults $c_j^{(b_j)}$'s corresponding to all the check line symbols $c_j^{b_j}$'s in Y_i are entered into Q_1 except those check lines with both $c_j^{b_j}$ and $c_j^{\bar{b}_j}$ in the Y_i.

(iii) If $c_j^{b_j} \varepsilon Y_i$ and c_j is a primary input line or a fan-out branch from a primary input line, the fault $c_j^{(b_j)}$ is weighted 1. If $c_j^{b_j} \varepsilon Y_i$ and c_j is a fan-out branch not from a primary input line, the fault $c_j^{(b_j)}$ is weighted with the total number of input variables or their complements in X_i on which $c_j^{b_j}$ depends. The weight of a fault $\alpha = (c_{j1}^{(b_{j1})}, \ldots, c_{jr}^{(b_{jr})})$ can be calculated with the total weights of the check line symbols in Y_i that α depends on. Weight each fault in Q_J.

Now, we can present the algorithm for finding $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ for Type 1 circuits as follows:

C-1) Set $J = 1$ and $L =$ an empty list.

C-2) Construct a Jth table and discard those rows which have non-zero entries under the I columns, that correspond to any I-term-less functions of F_c, which is constructed from F_c by removing the I terms corresponding to the non-zero entries in a row α_{ji} in the Jth table and substituting all the remaining Y_k's by 1, in the list L. When $J = 1$, none of its rows are discarded. If a Jth table is unable to be constructed because no α_{ji} is qualified, then the L list is $\mathcal{N}_1^*$, and go to Step C-4). Otherwise go to Step C-3).

C-3) Start with the lowest weighted row α_{ji}. Form an I-term-less fault function $F\alpha_{ji}^*$. Check if $F\alpha_{ji}^* = F$ or not. If not, enter $F\alpha_{ji}^*$ into L and cross out the row α_{ji} and other rows that have non-zero entries under the same columns as the row α_{ji}. Continue this process with another unchecked lowest weighted row in the table. If yes, leave the row α_{ji} in the table and continue this process with another unchecked lowest weighted row. After exhausting (crossing or checking) all the weighted rows in the table, set $J = J+1$ and go to Step C-2).

C-4) Set $J = 1$ and $M =$ an empty list.

C-5) For a term $X_i Y_i$ in F_c, construct a list Q_J. If $Q_J = \emptyset$, go to Step C-7). Otherwise, go to Step C-6).

C-6) Starting from the lowest weighted fault α_{jm} in Q_J, calculate $(X_i)\alpha_{jm}$.

If $(X_i)\alpha_{jm}\cdot\overline{F} \neq 0$, enter $(X_i)\alpha_{jm}$ into a list M_i and cross out from Q_J the fault α_{jm} and each of the faults satisfying the condition that every component of α_{jm} affects at least one of the check lines represented by one component of the fault. Continue this process with another lowest weighted unchecked fault in Q_J. If $(X_i)\alpha_{jm}\cdot\overline{F} = 0$, leave α_{jm} in Q_J and continue this process with another unchecked lowest weighted fault in Q_J. When all the faults in Q_J are crossed out or checked, set $J = J + 1$ and go to step C-5).

C-7) If Steps C-5) and C-6) have been applied to all the terms in F_c, go to Step C-8). Otherwise set $J = 1$ and go to Step C-5) for another term of F_c.

C-8) Find $M = \bigcup_i M_i$, which is $\mathcal{X}_2^*$, and then STOP.

In Step C-6) of Algorithm C, we check $(X_i)\alpha_{jm}\cdot\overline{F}$ to see if it equals 0 or not. If $(X_i)\alpha_{jm}\cdot\overline{F} = 0$, $(X_i)\alpha_{jm}$ must be equal to X_i for a Type 1 logic circuit, but $(X_i)\alpha_{jm}$ may have the relation that $X_i \subset (X_i)\alpha_{jm} \subseteq F$ for a Type 2 logic circuit. Hence a simple further check in Step C-6) to see if $(X_i)\alpha_{jm} = X_i$ or not can determine the type of the circuit and obtain the possible $(X_i)\alpha_{jm}$'s such that $X_i \subset (X_i)\alpha_{jm} \subseteq F$ if the circuit is of Type 2. Let G be a table produced by the modified Step C-6) in Algorithm C, which contains all the $(X_i)\alpha_{jm}$'s where $X_i \subset (X_i)\alpha_{jm} \subseteq F$, under the corresponding columns X_i's. If G is empty, the circuit is of Type 1 and the L and M produced by Algorithm C with modified Step C-6) is the $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$ of the circuit. If G is not empty, then the circuit is of Type 2. In this case, M still is $\mathcal{X}_2^*$, but L is not $\mathcal{X}_1^*$. We can, however, find $\mathcal{X}_1^*$ from L as follows:

Let $L = \{F_1^*,\ldots,F_n^*\}$ and $\{F_{ij}^*\}$ denote the set of functions generated from F_i^* by substituting all the possible combinations of $(X_r)\alpha_{jm}$'s, which appear in the table G under the terms X_r's, into the corresponding terms in F_i^* such that $F_i^* \subset F_{ij}^* \subset F$ and $F_{im}^* \not\subset F_{in}^*$ for different m and n. Hence, we obtain a new set L' from L, where $L' = \{\ \{F_{1j}^*\},\ldots,\{F_{nj}^*\}\ \}$ can be shown to be the representative set $\mathcal{X}_1^*$ for a Type 2 logic circuit.

V. EXAMPLES

Example 1. Consider the logic circuit shown in Fig. 1. Use Algorithm B to find the check function F_c. The lines have already been labeled according to Step B-1). Step B-2) yields

$$\begin{aligned} F_c &= d_5 = d_3 + d_4 = d_1c_6 + \bar{c}_7\bar{d}_2 \\ &= \bar{c}_1(\bar{c}_2 + \bar{c}_3)c_6 + c_2c_3\bar{c}_7(\bar{c}_4 + \bar{c}_5) \\ &= \bar{c}_1(\bar{c}_2c_6) + \bar{c}_1(\bar{c}_3c_6) + (c_2c_3\bar{c}_7)\bar{c}_4 + (c_2c_3\bar{c}_7)\bar{c}_5. \end{aligned}$$

Step B-3) yields the check function

$$\begin{aligned} F_c &= \bar{x}_1\bar{x}_2\bar{c}_1(\bar{c}_2c_6) + \bar{x}_1\bar{x}_3\bar{c}_1(\bar{c}_3c_6) + x_2x_3\bar{x}_3(c_2c_3\bar{c}_7)\bar{c}_4 + x_2x_3\bar{x}_4(c_2c_3\bar{c}_7)\bar{c}_5 \\ &= \sum_{k=1}^{4} X_kY_k \end{aligned}$$

Note that, for the purpose of easy recognition, the fan-out check line symbol and those check line symbols it depends on are grouped together in each term.

Now, Algorithm C is used to generate $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$. From Steps C-1) and C-2) we have the 1st table.

	X_1	X_2	X_3	X_4	weight
	$(x_1\bar{x}_2)$	$(\bar{x}_1\bar{x}_3)$	$(x_2x_3\bar{x}_3)$	$(x_2x_3\bar{x}_4)$	
$c_1^{(1)}$	1	1	0	0	2
$c_2^{(0)}$	1*	0	1	1	-
$c_2^{(1)}$	1	0	0	1*	-
$c_3^{(0)}$	0	1*	1	1	-
$c_3^{(1)}$	0	1	1*	1*	-
$c_4^{(1)}$	0	0	1	0	1
$c_5^{(1)}$	0	0	0	1	1
$c_6^{(0)}$	1	1	0	0	2
$c_7^{(1)}$	0	0	1	1	2

In Step C-3) start from the row with the smallest weight, row $c_4^{(1)}$. Since $X_1 + X_2 + X_4 = F$, we leave $c_4^{(1)}$ in the table and check $c_5^{(1)}$. Since $F^*_{c_5}(1) = X_1 + X_2 + X_3 \neq F$, $F_{c_5}(1)$ is entered into L, and rows $c_5^{(1)}$, $c_7^{(1)}$, $c_2^{(0)}$, $c_2^{(1)}$, $c_3^{(0)}$ and $c_3^{(1)}$ are crossed out. Now we check row $c_1^{(1)}$ of weight 2. Since $F^*_{c_1}(1) = X_3 + X_4 \neq F$, it is entered into L and $c_1^{(1)}$, $c_6^{(0)}$ are crossed out. Since all rows are crossed out except $c_4^{(1)}$, the 2nd table has to be empty. Hence, we obtain $L = \{\bar{x}_1\bar{x}_2 + \bar{x}_1\bar{x}_3,\ x_2x_3\bar{x}_4\}$ and go to Step C-4). In Steps C-5) and C-6) for $X_1Y_1 = \bar{x}_1\bar{x}_2\bar{c}_1(\bar{c}_2c_6)$, $Q_1 = \{c_1^{(0)}|1,\ c_2^{(0)}|1,\ c_6^{(1)}|1\}$, where the number attached to each fault indicates the weight of the fault. Now, $(X_1)_{c_1}(0) = \bar{x}_2$ and $(X_1)_{c_2}(0) = \bar{x}_1$, where $\bar{x}_2\cdot\bar{F} \neq 0$ and $\bar{x}_1\cdot\bar{F} \neq 0$. So $M_1 = \{\bar{x}_1, \bar{x}_2\}$ and all the faults in Q_1 are crossed out. Similarly, $M_2 = \{\bar{x}_1, \bar{x}_3\}$, $M_3 = \{x_2\bar{x}_3,\ x_2x_3\}$ and $M_4 = \{x_2x_3,\ x_2\bar{x}_4,\ x_3\bar{x}_4\}$. Hence in Step C-8) $M = \{\bar{x}_1,\ \bar{x}_2,\ \bar{x}_3,\ x_2x_3,\ x_2\bar{x}_3,\ x_2\bar{x}_4,\ x_3\bar{x}_4\}$. Considering the function sets L and M as $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$, we now apply Algorithm A to obtain the test set. In Step A-2), one of the minimum covers of $(\overline{H_{1i}{}^*\cdot F})$'s is found to be $\{x_1x_2x_3\bar{x}_4,\ \bar{x}_1\bar{x}_2x_3\bar{x}_4\}$ and one of the minimum covers of $(H_{2i}{}^*\cdot\bar{F})$'s is $\{x_1x_2\bar{x}_3x_4,\ x_1\bar{x}_2\bar{x}_3x_4,\ \bar{x}_1x_2x_3\bar{x}_4\}$. Hence in Step A-3), we obtain a test set of 5 tests $T = \{1110/1,\ 0010/1,\ 1101/0,\ 1001/0,\ 0110/0\}$, where the 0 or 1 attached to each input test pattern indicates the response of the circuit without any fault when the input test pattern is applied.

Example 2. Consider the highly redundant logic circuit shown in Figure 2, which has also been considered by Bossen and Hong [5]. The check function is

$$\begin{aligned} F_c &= \bar{y}wxyz\ \bar{c}_1c_2c_3c_4c_5 + \bar{y}\bar{z}yA\ \bar{c}_1(\bar{c}_6c_7c_8c_{10}) \\ &\quad + \bar{z}yA(\bar{c}_6c_7c_8c_{11}) + A\ c_9 \\ &= \sum_{k=1}^{4} X_kY_k \end{aligned}$$

After Step C-6), we find that

$$L = \{F_1^* = X_1 + X_2 + X_3\},$$

and the table G is

X_2 $(\bar{y}\bar{z}yA)$	X_3 $(\bar{z}yA)$
$A\bar{y}\bar{z}$	Ay
$Ay\bar{z}$	$A\bar{z}$
Ay	
$A\bar{z}$	
$A\bar{y}$	

Since G is not empty, this circuit is of Type 2. Thus, we have $\{F_{1j}^*\} = \{Ay + A\bar{z}, A\bar{y} + A\bar{z}\}$. For L contains F_1^* only, we obtain $\mathcal{X}_1^* = \{\ \{F_{1j}^*\}\ \} = \{Ay + A\bar{z}, A\bar{y} + A\bar{z}\}$. Because there are only two elements in $\mathcal{X}_1^*$, at most two tests are obtained from $\mathcal{X}_1^*$.

VI. CONCLUSIONS

An algorithm for generating multiple fault-detection test set has been presented in this paper. Although we treat the logic circuits of Type 1 and Type 2 differently, since we can identify the type of a given circuit in the algorithm, the generality of the algorithm is preserved. In our algorithm, for a fault function F_i^* which can be detected by the input test patterns represented by the minterms in both F and $\bar{F}$, where F is an output function of a logic circuit, we classify it as an element in $\mathcal{S}_2^*$, and thus during the process of test generation we only consider those input test patterns represented by the minterms in $\bar{F}$. Hence, a near minimum test is obtained instead of an optimal one. In this sense, Bossen and Hong's general algorithm not only considers F_i^* in the process of generating the so-called abnormal true tests, but also reconsiders F_i^* in the process of generating the so-called abnormal false tests. Since it is not necessary to consider F_i^* twice during the generation of tests, this is the main reason why Bossen and Hong's general algorithm always generates more tests. Furthermore, because the effect of a fault to a logic circuit has been deeply incorporated in this algorithm, we can find a smaller set of representative fault functions which are sufficient to generate a test set containing fewer tests than those generated by other existing methods. Comparing this Algorithm to that of Bossen and Hong [5], we can find that (a) the cause-effect equations for both F and $\bar{F}$ have to be derived in their algorithm, and (b) there are more elements in sets X_m and X_j in their algorithm than those in $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$, and hence their algorithm is more complicated for finding the test set. Examples 1 and 2 have also been considered with Bossen and Hong's algorithm. However, for Example 1, a test set of at least 6 tests was generated by Bossen and Hong's general method, while we can find a set of 5 tests, which actually is an optimal test set. In example 2, two tests are obtained from $\mathcal{X}_1$* in our algorithm, while Bossen and Hong's algorithm generated 3 such tests.

References:

[1] J. M. Galey, R. E. Norby and J. P. Roth, "Techniques for the diagnosis of switching circuits failures," IEEE Trans. on Communication and Electronics, Vol. 83, pp. 509-514, Sept. 1964.

[2] A. D. Friedman, "Fault detection in redundant circuits," IEEE Trans. on Electronic Computers, Vol. EC-16, pp. 99-100, Feb. 1967.

[3] S. S. Yau and Y. S. Tang, "An efficient algorithm for generating complete test sets for combinational logic circuits," IEEE Trans. on Computers, Vol. C-20, No. 11, pp. 1245-1251, Nov. 1971.

[4] A. R. Klayton and A. K. Susskind, "Multiple-fault-detection tests for loop-free logic networks," Dig. 1971 IEEE Int. Comput. Soc. Conf., pp. 77-78.

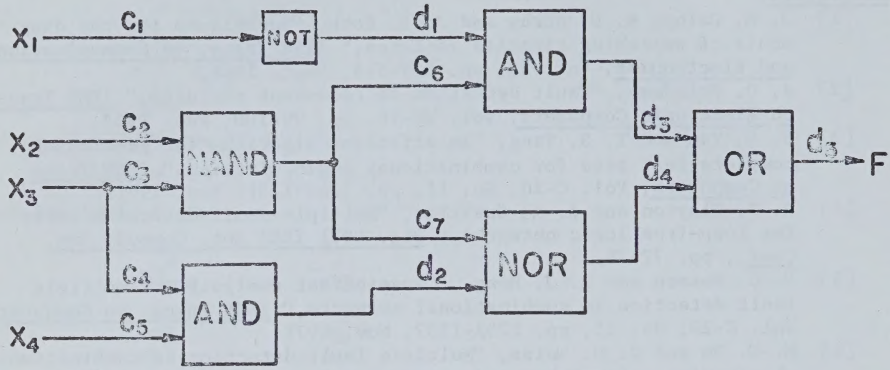
[5] D. C. Bossen and S. J. Hong, "Cause-effect analysis for multiple fault detection in combinational networks," IEEE Trans. on Computers, Vol. C-20, No. 11, pp. 1252-1257, Nov. 1971.

[6] M.-W. Du and C. D. Weiss, "Multiple fault-detection in combinational circuits: algorithms and experimental results," Dig. 1972 Int. Symposium on Fault Tolerant Computing, pp. 120-125, June 1972.

[7] I. Kohavi and Z. Kohavi, "Detection of multiple faults in combinational logic networks," IEEE Trans on Computers, Vol. C-21, No. 6, pp. 556-568, June, 1972.

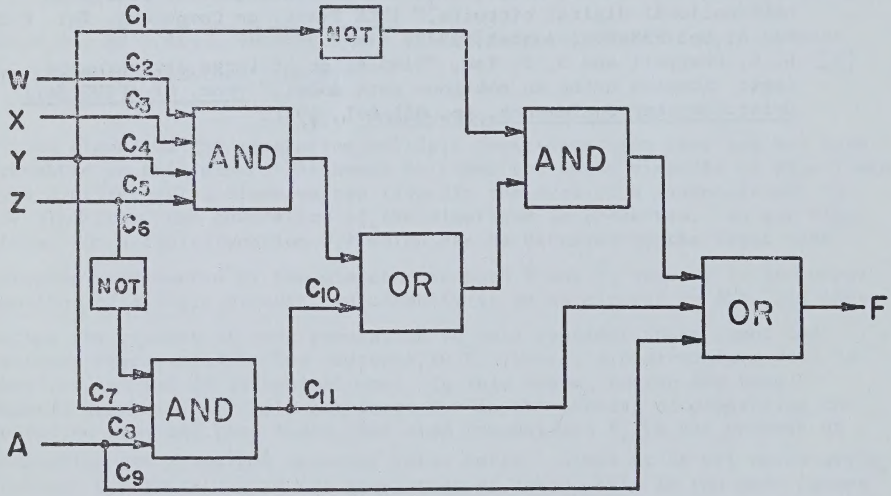
[8] D. R. Schertz and G. Metze, "A new representation for faults in combinational digital circuits," IEEE Trans. on Computers, Vol. C-21, No. 8, pp. 858-866, August, 1972.

[9] S. G. Chappell and S. S. Yau, "Simulation of large asynchronous logic circuits using an ambigous gate model," Proc. of AFIPS Fall Joint Computer Conference, pp. 651-661, 1971.



X1
C1
NOT
d1
AND
C6
X2
C2
NAND
X3
C3
d5
OR
d3
F
d4
C7
NOR
C4
AND
d2
X4
C5

Figure 1.



C1
NOT
W
C2
X
C3
C4
Y
AND
AND
C5
Z
C6
NOT
OR
C10
OR
F
C7
AND
C11
A
C8
C9

Figure 2.

DETECTION OF INTERMITTENT FAULTS IN COMBINATIONAL CIRCUITS

SAMIR KAMAL
Wayne State University
Detroit, Michigan

CARL V. PAGE
Michigan State University
East Lansing, Michigan

ABSTRACT

Intermittent faults are those faults whose effect on the behavior of the circuit is present only part of the time. A probabilistic model is proposed for these faults. A detection procedure, employing the repetition of tests that detect these faults had their effect been permanent, is proposed. Optimization of the detection experiment is investigated.

INTRODUCTION

A circuit having an intermittent fault may show the effect of such a fault when an input test is applied one time, and possibly not show it when the same input test is applied other times. There are many factors that can result in the existence of intermittent faults such as : critical circuit timing, close design tolerances, noise, charges on stray capacitances, loose connections, aging, fatigue, or irregular physical structure of components [1,9].

Almost all of the work cited in the literature in the area of modelling, detection, and diagnosis of faults in logical circuits deals only with permanent faults [1,3,8]. Ad-hoc methods have been used to handle intermittent faults in practice, since a formal treatment has not been developed. In this paper, a model for intermittent faults is proposed, also a detection procedure is suggested. Optimization of the detection experiment is also investigated. The proofs to the theorems and to the corollaries are omitted due to space limitations; they are found in [7].

THE MODEL

The model proposed is a probabilistic one. The following assumptions are used:

(1) The circuit can have only one intermittent fault during testing (the single fault assumption).

(2) The circuit is irredundant; thus the effect of a fault would not be masked.

(3) The faults are well behaved, i.e., during the application of a test, the circuit behaves as if either it is fault free, or else possesses a permanent fault for that period of time.

(4) The faults are signal independant, i.e., the presence of a fault does not depend on the signal values present in the circuit.

The notation used in this paper is close to that employed in pattern recognition literature, for example, see [4]. The basic elements of the model are defined below.

Intermittent Fault Space Ω : $\Omega = \{\omega_0, \omega_1, \ldots, \omega_n\}$

Ω is the set of intermittent faults of interest that the circuit might have,

ω_0 : denotes the fault free circuit condition,

ω_i : $(1 \leq i \leq n)$ denotes the condition of the circuit having intermittent fault number i.

The condition of the circuit can be described by a single element from Ω, i.e., the circuit is either fault free or it has only one of n possible intermittent faults (the single fault assumption). During the testing experiment, the circuit will have the same intermittent fault condition. If fault condition ω_i exists in the circuit, it does not mean that its effect on the

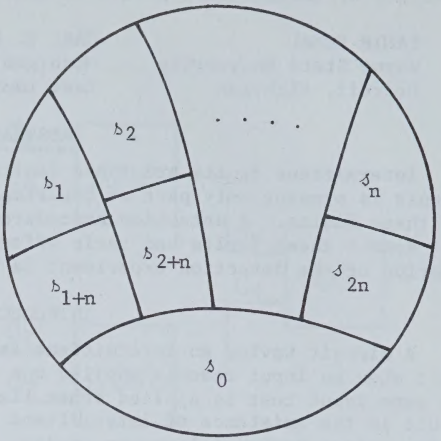


Figure 1. Action Space S.

circuit behavior will be present all the time; this is due to the intermittency of the faults.

Permanent Fault Space F : $F = \{f_0, f_1, \ldots, f_n\}$

There is one to one correspondence between F and Ω. f_i, $(1 \leq i \leq n)$, is the permanent fault that the circuit would have if the effect of ω_i is present all the time. f_0 corresponds to the fault free condition.

Action Space S : $S = \{s_0, s_1, \ldots, s_{2n}\}$

Every point in represents an instantaneous state condition of the circuit as follows :

s_0 : denotes the state of being fault free,

s_i : $(1 \leq i \leq n)$ denotes the state of the circuit when it possesses intermittent fault ω_i and the effect of this fault is present,

s_{i+n} : $(1 \leq i \leq n)$ denotes the state of the circuit when it possesses intermittent fault ω_i and the effect of this fault is not present.

The elements of Ω define events on S as follows :

$$\omega_0 = \{s_0\}$$
$$\omega_i = \{s_i, s_{i+n}\} \quad (1 \leq i \leq n). \tag{1}$$

Due to the single fault assumption, during testing, the instantaneous state of the circuit will either stay as s_0 or change randomly between s_i and s_{i+n} for some, but fixed, i $(1 \leq i \leq n)$. A schematic diagram for S is shown in Figure 1.

Test Set τ : $\tau = \{t_1, t_2, \ldots, t_m\}$

τ is a complete test set that detect all faults in F. It can be generated using any of the methods employed to generate tests that detect permanent faults. In general, a test will detect more that one fault. τ_i will be used to denote the subset of τ that contains all tests detecting f_i.

Random Variable Set T : $T = \{T_1, T_2, \ldots, T_m\}$

Every test t_j $(1 \leq j \leq m)$ in τ defines a random variable T_j on S. If test t_j is applied to the circuit and it fails, i.e., the observed output is different from that of the normal (fault free) circuit, the value of T_j is defined to be 0. On the other hand if t_j does not fail, i.e., produces an output identical to that of the normal circuit, then the value of T_j is defined to be 1. Formally, T_j is a function from S into the set $\{0,1\}$ defined as :

$$T_j : S \rightarrow \{0,1\} \quad , \quad (1 \leq j \leq m) \; ;$$
$$T_j(s_i) = \begin{cases} 0 & 1 \leq i \leq n, \text{ and } t_j \text{ detects } f_i \, . \\ 1 & \text{otherwise} \, . \end{cases} \tag{2}$$

T_i will be used to denote the subset of T that corresponds to τ_i.

k-Dimensional Random Variable Set T^k :

T^k is the cartesian product of T by itself k times. An element of T^k is a k-tuple of random variables from T. The value of such an element is a k-dimensional binary vector which corresponds to the outcome of an experiment with k tests. The notation $b = \vec{1}^k$ will be used to denote a k-dimensional binary vector, all of its entries are 1's. This corresponds to the outcome of an experiment of k tests; none of them has failed.

Prior Probabilities $P(\omega_i)$: $(0 \leq i \leq n)$

$p_i = P(\omega_i)$, $0 \leq i \leq n$, is the prior probability that the circuit is in state ω_i. The values of these p_i's are assumed to be known in the model. These values could be obtained empirically, from manufacturer information, or from experience.

Conditional Probability of Malfunction $P(\delta_i/\omega_i)$: $(1 \leq i \leq n)$

$e_i = P(\delta_i/\omega_i)$, $1 \leq i \leq n$, is the probability that the effect of intermittent fault ω_i will be present knowing that the circuit already has intermittent fault ω_i. The values of e_i's, assumed to be known in the model, are constant during the course of the experiment.

DETECTION OF INTERMITTENT FAULTS

In general, fault detection means applying tests from a certain test set to find out whether a given circuit is fault free or not. To detect permanent faults, any particular test need only be applied once. The approach proposed for detection of intermittent faults is through repeated application of tests from the test set τ. The repetition of a particular test is needed due to the fact that the circuit might have an intermittent fault that could be detected by this test but the effect of such a fault is not present during the first time this test is applied. After repeated application of a particular test and obtaining outputs identical with those of a fault free circuit, we can reason that the circuit might still have an intermittent failure whose effect has not yet been observed. Should the test be repeated forever? What is needed is a "wise" stopping rule. It is assumed that testing is done by a fast machine, say a computer, so repetition, possibly in the order of millions of times, can be done fast and fairly easily. The problem of detection of intermittent faults can then be viewed as a statistical decision problem. An appropriate decision rule is sequential in nature. The ones suggested make use of the posterior probabilities.

Posterior Probabilities $P(\omega_i/T_j)$

After applying a test t to the circuit and observing the output, the probabilities $P(\omega_i/T_j)$, $0 \leq i \leq n$, can be calculated using Bayes' Rule:

$$P(\omega_i/T_j) = \frac{P(T_j/\omega_i)\,P(\omega_i)}{\sum_{k=0}^{n} P(T_j/\omega_k)P(\omega_k)} . \tag{3}$$

As the prior probabilities $P(\omega_i)$, $0 \leq i \leq n$, reflect the beliefs about the condition of the circuit before any test is applied, the posterior probabilities $P(\omega_i/T_j)$ are an up-dated version of these beliefs after applying test t and observing the output. Of course, the values of these posterior probabilities depend on the observed output, i.e., on the value of T_j. These posterior probabilities will be used as prior probabilities next time a test is applied. We suggest the following detection procedure.

Detection Procedure

(1) Apply an appropriate test t_j from τ. The meaning of "appropriate" should be clear after defining the decision rule. If the output of the circuit is different from that of the fault free circuit (i.e., if $T_j = 0$) decide that the circuit has an intermittent fault and stop, otherwise go to (2).

(2) At this point, the output so far is identical to that of a fault free circuit. Calculate the posterior probabilities $P(\omega_i/T_j = 1)$, $0 \leq i \leq n$, using Bayes' Rule.

(3) Using the posterior probabilities apply a decision rule, and decide one of two things:
   (a) Request the application of a particular test and repeat, i.e., go to (1), or,
   (b) Decide that the circuit is fault free and stop.

Decision Rule

The decision rule mentioned above can be freely selected to satisfy a set of conditions that represent acceptable measures for deciding that the circuit is fault free. Moreover, the decision rule must insure the termination of the detection procedure in a finite amount of time, i.e., insure that we have a detection algorithm. Examples of typical decision rules are presented later.

A Simple Case

Let $\Omega = \{\omega_0, \omega_1\}$, i.e., one particular intermittent fault could possibly exist in the circuit.

Let $\tau = \{t_1\}$. It follows that:

$S = \{s_0, s_1, s_2\}$
$F = \{f_0, f_1\}$
$\omega_0 = \{s_0\}$
$\omega_1 = \{s_1, s_2\}$
$T_1(s_0) = T_1(s_2) = 1$
$T_1(s_1) = 0$

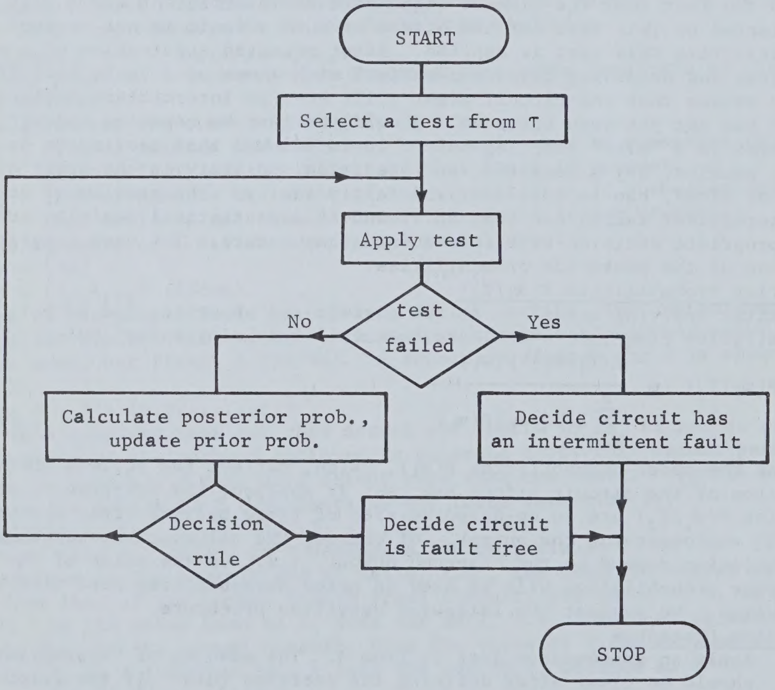

START
Select a test from τ
Apply test
test failed
No
Yes
Calculate posterior prob., update prior prob.
Decide circuit has an intermittent fault
Decision rule
Decide circuit is fault free
STOP

Figure 2. Flow Chart for Detection Procedure.

Consider the following distribution on $\mathcal{S}$:

$P(\omega_1) = p_1$

$P(\omega_0) = p_0 = 1 - p_1$

$P(\mathcal{S}_1/\omega_1) = e_1$

Using (3) then,

$$p_{1,1} = P(\omega_1/T_1 = 1) = \frac{P(T_1 = 1/\omega_1)\ P(\omega_1)}{P(T_1 = 1/\omega_0)\ P(\omega_0) + P(T_1 = 1/\omega_1)\ P(\omega_1)}$$

$$= \frac{(1-e_1)\ p_1}{1 - e_1\ p_1} \tag{4}$$

$(1-e_1)$ is less than $(1-e_1\ p_1)$, therefore:

$$p_{1,1} < p_1 \tag{5}$$

Similarly, $p_{0,1} = P(\omega_0/T_1 = 1) = \dfrac{p_0}{1-e_1\ p_1}$ (6)

it follows that:

$$p_{0,1} > p_0 \tag{7}$$

$$P(\omega_1/T_1 = 0) = 1 \tag{8}$$

Similarly, $P(\omega_0/T_1 = 0) = 0$ (9)

The interpretation of (5) and (7) is that, if t_1 is applied and the circuit produced a correct output (identical with that of a fault free circuit), then our certainty about the circuit being faulty will decrease and our certainty about the circuit being fault free will increase; which is quite reasonable. On the other hand, the interpretation of (8) and (9) is that, once a wrong (different from that of a fault free circuit) output is observed upon the application of t_1, then we know for sure that the circuit has an intermittent fault.

The posterior probabilities after applying t_1 for k+1 times and observing k+1 correct outputs are (using (3)):

$$p_{1,k+1} = P(\omega_1/T_1=1, b=\overrightarrow{1^k}\ ;\ b \in \Upsilon^k) = \frac{(1-e_1)\ p_{1,k}}{1-e_1\ p_{1,k}} \tag{10}$$

Similarly, $p_{0,k+1} = P(\omega_0/T_1=1, b=\overrightarrow{1^k}\ ;\ b \in \Upsilon^k) = \dfrac{p_{0,k}}{1 - e_1\ (1-p_{0,k})}$ (11)

From (10) and (11) it follows that:

$$p_{1,k+1} < p_{1,k} \tag{12}$$

$$p_{0,k+1} > p_{0,k} \tag{13}$$

Theorem 1

$$\text{(a)} \quad p_{1,k} = \frac{(1-e_1)^k\ p_1}{p_0 + (1-e_1)^k\ p_1} \tag{14}$$

$$\text{(b)} \quad p_{0,k} = \frac{p_0}{p_0 + (1-e_1)^k\ p_1} \tag{15}$$

Corollary 1

$$\text{(a)} \quad \lim_{k\to\infty} p_{1,k} = 0 \tag{16}$$

$$\text{(b)} \quad \lim_{k\to\infty} p_{0,k} = 1 \tag{17}$$

Definition 1

The likelihood ratio λ_k is defined as:

$$\lambda_k = \frac{p_{1,k}}{p_{0,k}} \quad . \tag{18}$$

From (14) and (15), it follows that:

$$\lambda_k = \frac{(1-e_1)^k \ p_1}{(1-p_1)} \tag{19}$$

Corollary 2

$$\lambda_{k+1} < \lambda_k \tag{20}$$

Corollary 3

$$\lim_{k\to\infty} \lambda_k = 0 \tag{21}$$

A Decision Rule Employing the Posterior Probability. Earlier it was stated that the posterior probabilities will be used in the decision rule to decide whether to continue testing or that the circuit is fault free. The following decision rule is suggested to be part of the detection procedure:

> If the posterior probability $p_{1,k}$ goes below a certain threshold s ($0<s<1$), decide that the circuit is fault free and stop, otherwise apply t_1 and repeat (i.e., go to step (1) of the detection procedure).

This decision rule is acceptable, since it guarantees termination of the detection procedure in a finite amount of time because $p_{1,k}$ is monotonically decreasing (from (12)), and as k increases it can get below the threshold s (from (16)).

The value of the threshold s can be chosen by considering many factors such as the length of the experiment, and the criticalness of the proper operation of the circuit.

To determine a least upper bound on the length of the experiment, find k for which $p_{1,k} < s$. From (14):

$$\frac{(1-e_1)^k \ p_1}{(1-p_1) + p_1 \ (1-e_1)^k} < s$$

which gives:

$$k > \frac{\log \dfrac{s \ (1-p_1)}{p_1 \ (1-s)}}{\log \ (1-e_1)} \tag{22}$$

Another Decision Rule Employing the Likelihood Ratio. The rule suggested here compares the likelihood ratio (which is a function of the posterior probabilities) with a threshold u ($u>0$) as follows:

> If λ_k goes below u decide that the circuit is fault free and stop, otherwise apply t_1 and repeat (i.e., go to step (1) of the detection procedure).

This decision rule is also acceptable, since it guarantees termination of the detection procedure in a finite amount of time because λ_k is monotonically decreasing (from (20)), and as k increases it can get below u (from (21)).

The value of the threshold u could be chosen in a fashion similar to that of selecting s.

The least upper bound on the length of the testing experiment is obtained by finding k such that $\lambda_k<u$. From (19) we obtain:

$$k > \frac{\log \dfrac{u \ (1-p_1)}{p_1}}{\log \ (1-e_1)} \tag{23}$$

The General Case

This is the case described by the model proposed earlier.

Consider the probability distribution on S governed by the following conditions:

$$P(\omega_i) = p_i \quad , \quad 0 \leq i \leq n$$

$$P(\delta_i/\omega_i) = e_i \quad , \quad 1 \leq i \leq n$$

Definition 2

The membership functions M_j $(1 \leq j \leq m)$ from the set $\{0,1,\dots,n\}$ into the set $\{0,1\}$ are defined as:

$$M_j(0) = 0 \quad ,$$

$$M_j(i) = \begin{cases} 1 & \text{if } t_j \text{ detects } f_i \text{ .} \\ 0 & \text{if } t_j \text{ does not detect } f_i \text{ .} \end{cases} \qquad (1 \leq i \leq n)$$

The posterior probabilities are now calculated using Bayes' Rule. Note that:

$$P(T_j = 0/\omega_i) = \begin{cases} e_i & \text{if } t_j \text{ detects } f_i \text{ .} \\ 0 & \text{if } t_j \text{ does not detect } f_i \text{ .} \end{cases}$$

$$\text{and, } P(T_j = 1/\omega_i) = \begin{cases} 1-e_i & \text{if } t_j \text{ detects } f_i \text{ .} \\ 1 & \text{if } t_j \text{ does not detect } f_i \text{ .} \end{cases}$$

In terms of the membership functions, we can write:

$$P(T_j = 0/\omega_i) = e_i \; M_j(i) \text{ .} \tag{24}$$

$$P(T_j = 1/\omega_i) = 1 - e_i \; M_j(i) \text{ .} \tag{25}$$

$$\text{Thus,} \quad P(\omega_i/T_j = 0) = \frac{e_i \; M_j(i) \; P(\omega_i)}{\sum_{\ell=0}^{n} e_\ell \; M_j(\ell) \; P(\omega_\ell)} \tag{26}$$

It follows that:

$$P(\omega_0/T_j = 0) = 0 \text{ .} \tag{27}$$

If t_j does not detect f_i, then:

$$P(\omega_i/T_j = 0) = 0 \text{ .} \tag{28}$$

If t_j detects f_i, then:

$$P(\omega_i/T_j = 0) = \frac{e_i \; P(\omega_i)}{\sum_{\ell=0}^{n} e_\ell \; M_j(\ell) \; P(\omega_\ell)} \text{ .} \tag{29}$$

Equations (27), (28) and (29) give the value of the posterior probabilities if the applied test t_j fails.

If t_j does not fail, it can be shown that:

$$P(\omega_0/T_j = 1) > P(\omega_0) \text{ .} \tag{30}$$

If t_j does not detect f_i, then:

$$P(\omega_i/T_j = 1) > P(\omega_i) \text{ .} \tag{31}$$

This is an interesting result, since it indicates that the certainty about the circuit having a particular fault increases (if the applied test does not detect that fault) even though the applied test produced a correct output (that is, identical with that of a fault free circuit).

If t_j detects f_i, $P(\omega_i/T_j = 1)$ could be less than, equal to, or even greater than $P(\omega_i)$, i.e.,

$$P(\omega_i/T_j = 1) \gtreqless P(\omega_i) \tag{32}$$

This is a more interesting result since the certainty about the circuit having a particular fault could increase even if the applied test detects that fault and produces a correct output.

From (31) and (32), it is clear that a decision rule based on comparing the posterior probabilities with some thresholds, as was done for the simple case, is not acceptable since the posterior probabilities are not monotonically decreasing functions.

The posterior probabilities after applying k+1 tests; the k+1-st being, say, test t ; are:

$$P(\omega_i/T_j = 1,\ b\ ;\ b \in \Upsilon^k) = \frac{[1-e_i\ M_j(i)]\ P(\omega_i/b)}{\sum_{\ell=0}^{n} P(T_j = 1/\omega_\ell)\ P(\omega_\ell/b)} \tag{33}$$

Definition 3

If k+1 tests from τ are applied; the k+1-st being, say, test t_j; and a correct output was observed every time, the likelihood ratios $\lambda_{i,k+1}$ $(1 \le i \le n)$ are defined as:

$$\lambda_{i,k+1} = \frac{P(\omega_i/T_j = 1,\ b = \overrightarrow{1^k}\ ;\ b \in \Upsilon^k)}{P(\omega_0/T_j = 1,\ b = \overrightarrow{1^k}\ ;\ b \in \Upsilon^k)}\ . \tag{34}$$

From (33), it follows that:

$$\lambda_{i,k+1} = [1-e_i\ M_j(i)]\ \lambda_{i,k}\ . \tag{35}$$

Thus, $\lambda_{i,k+1} < \lambda_{i,k}$. (36)

The equality in (36) holds only if the k+1-st test detects f_i, otherwise this relation is strict inequality.

Definition 4

The initial likelihood ratios λ_i $(1 \le i \le n)$ are defined as:

$$\lambda_i = \lambda_{i,0} = \frac{P(\omega_i)}{P(\omega_0)}\ . \tag{37}$$

Theorem 2

If k tests from τ are applied, ℓ of them are from τ_i, and none of them failed, then:

$$\lambda_{i,k} = (1-e_i)^{\ell}\ \lambda_i\ . \tag{38}$$

Corollary 4

$$\lim_{\ell \to \infty} \lambda_{i,k} = 0 \tag{39}$$

The Decision Rule. The decision rule suggested, compares the likelihood ratios $\lambda_{i,k}$, $(1 \le i \le n)$, with thresholds u_i ($u_i > 0$ for all i) as follows:

If $\lambda_{i,k} \le u_i$ for all i, decide that the circuit is fault free and stop.
If $\lambda_{i,k} > u_i$ for some i, select a test from τ_i and repeat, (i.e., go to step (1) of the detection procedure).

From (36) we see that the likelihood ratios are monotonically non-increasing functions. Any likelihood ratio $\lambda_{i,k}$ will strictly decrease if we apply a test from τ_i (and of course, that test produces a correct output). Thus, from (39), the likelihood ratios could go below the specified thresholds, and the detection procedure is guaranteed to terminate in a finite amount of time using this decision rule. Hence, it is an acceptable rule.

Theorem 2 can be used to determine a least upper bound on the number of tests k_i from τ_i ($1 \le i \le n$) that are needed, as follows:

$$(1-e_i)^{k_i} \lambda_i < u_i$$

$$\text{or,} \quad k_i > \frac{\log \frac{u_i \, p_0}{p_i}}{\log(1-e_i)} \quad . \qquad (40)$$

OPTIMUM DETECTION EXPERIMENT

Definition 5

The set of faults $\{f_1, f_2, \ldots, f_n\}$ together with the test set τ define the n x m fault table matrix $A = (a_{ij})$ as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } t_j \text{ detects } f_i \, . \\ 0 & \text{if } t_j \text{ does not detect } f_i \, . \end{cases} \qquad (41)$$

In the previous section, least upper bounds k_i's on the number of tests, that detect f_i, that are needed were obtained (40). However, a given test usually detects more than one fault; so, a method is needed to determine how many times every test should be repeated so that (40) is satisfied for all i ($1 \le i \le n$), and that the overall experiment length is minimum. If t_j ($1 \le j \le m$) is repeated x_j times, then, (40), in terms of the fault table matrix A (definition 5), yields,

$$\sum_{j=1}^{m} a_{ij} \, x_j \ge k_i \quad . \qquad (42)$$

The experiment length ℓ is:

$$\ell = \sum_{j=1}^{m} x_j \quad . \qquad (43)$$

ℓ is the function to be minimized.

The quantities at hand satisfy the following conditions:

1. $x_j \ge 0$
2. all x_j's are integers

This is an all integer-integer programming problem, the solution of which, determines the optimum number of repetitions for each test. For details and solutions of integer programming problems, see [5,6].

It should be noted that for the permanent faults case, every k_i ($1 \le i \le n$) will be equal to 1; thus, similar to the problem of minimizing boolean functions. Integer programming treatment for this special case is found in [2].

Example

Consider the following fault table matrix:

$$\begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ \hline f_1 & 1 & 1 & 0 & 1 & 0 \\ f_2 & 1 & 1 & 1 & 0 & 1 \\ f_3 & 0 & 1 & 1 & 1 & 1 \\ f_4 & 0 & 1 & 0 & 0 & 1 \\ f_5 & 0 & 0 & 0 & 1 & 1 \\ f_6 & 1 & 0 & 1 & 0 & 0 \end{array}$$

The corresponding integer programming problem is:

$$
\begin{array}{lllllll}
x_1 + x_2 & & + x_4 & & \geq & k_1 \\
x_1 + x_2 & + x_3 & & + x_5 & \geq & k_2 \\
\phantom{x_1 +{}} x_2 & + x_3 & + x_4 & + x_5 & \geq & k_3 \\
\phantom{x_1 +{}} x_2 & & & + x_5 & \geq & k_4 \\
 & & x_4 & + x_5 & \geq & k_5 \\
x_1 & + x_3 & & & \geq & k_6
\end{array}
$$

Find integer x 's that minimize $x_1 + x_2 + x_3 + x_4 + x_5$.

REFERENCES

[1] Chang, H. Y.; Manning, E.; and Metze G., *Fault Diagnosis of Digital Systems*. New York: Wiley-Interscience, 1970.

[2] Cobhan, A.; Fridshal, R.; and North, J. H., "An Application of Linear Programming to the Minimization of Boolean Functions," in *Proc. 2nd Annu. Symp. Switching Theory and Logical Design*, pp. 3-9, 1961.

[3] Friedman, A. D., and Menon, P. R., *Fault Detection in Digital Circuits*. Englewood Cliffs, N. J.: Prentice-Hall, 1971.

[4] Fu, K. S., *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic Press, 1968.

[5] Gomory, R. E., "All-Integer Integer Programming Algorithm," IBM Res. Report, RC-189, Jan. 1960.

[6] Hillier, F. S., and Lieberman, G. J., *Introduction to Operations Research*. San Francisco: Holden-Day, Inc., 1967.

[7] Kamal, S., "Diagnosis of Intermittent Faults in Digital Systems," Ph.D. thesis, Michigan State University, 1972.

[8] Roth, J. P., "Diagnosis of Automata Failures: A Calculus and a Method," *IBM J. Res. Dev.*, Vol. 10, pp. 278-291, July 1966.

[9] Yen, Y. T., "Intermittent Failure Problems of Four-Phase MOS Circuits," *IEEE J. Solid State Circuits*, Vol. SC-3, June 1969.

COMPLETE TEST SETS FOR LOGIC FUNCTIONS[1]

SUDHAKAR M. REDDY
Department of Electrical Engineering
University of Iowa
Iowa City, Iowa 52240

ABSTRACT

An earlier result due to Betancourt[1] is extended to multiple stuck-at-faults and arbitrary logic functions. The concept of an expanded truth table for logic function is introduced. It is proved that the set of minimal true vertices and maximal false vertices of the expanded truth table constitutes a "complete test set" to detect any number of stuck-at-faults in a network belonging to a class of restricted networks. It is further indicated that even in the presence of redundancies in the network, the test sets given remain valid.

I. INTRODUCTION

Solutions to the problem of designing test sets to detect faults in an arbitrary realization of a logic function are extremely important since normally, attempts to derive test sets for a given realization are unwieldy. Also the given realization may have redundancies in it which might cause the test sets to become invalid.[2] An elegant solution to this problem has been given by Betancourt[1] for the case of unate functions, where the networks realizing these functions are allowed to have a single permanent stuck-at-zero (s-a-0) or stuck-at-one (s-a-1) fault.

In this paper we extend Betancourt's[1] result to multiple stuck-at-faults and arbitrary logic functions. For the sake of completeness and ease in reading, we will give several definitions and develop a certain amount of notation in this section.

Notation:

f an n-variable logic function.

F is the network realizing f that is being tested for faults.

u is a unate function.

$\underline{X},\underline{Y}$ etc., are input vectors (vertices).

$\underline{X}$ is a true vertex of f if $f(\underline{X}) = 1$.

$\underline{X}$ is a false vertex of f if $f(\underline{X}) = 0$.

x_i is an input variable.

x_i^* is an input literal, which is x_i or $\overline{x}_i$ (complement of x_i).

---

[1] The research reported has been supported by ONR Grant #N00014-68-A-0500.

Definition 1: A logic function f is positive unate in literal x_i^* if for every input vector $\underline{X}$, such that $f(\underline{X}) = 1$, with x_i taking the value to make $x_i^* = 0$, also implies $f(\underline{Y}) = 1$ where $\underline{Y}$ is obtained by complimenting the value of x_i in $\underline{X}$. If a function is not positive unate in x_i or $\overline{x}_i$, then it is not unate in x_i. If f is positive unate in x_i^*, then f is not positive unate in $\overline{x}_i^*$.

Definition 2: A logic function f is a positive unate function of literals $x_{i_1}^*, x_{i_2}^*, \ldots, x_{i_\ell}^*$ if it is positive unate in $x_{i_j}^*$, $j \leq 1 \leq \ell$. Notice that this definition of a positive unate function differs from the usual definition, which is given in terms of input variables.

Definition 3: A logic network constructed from AND, OR, NAND, NOR gates is called a gate network. If only AND and/or OR gates are used and invertors, if necessary, are used at the external inputs to gates, then it is called a restricted gate network.

Definition 4: If the number of inversions in any path connecting two points in a gate network are the same, then the gate network is called a unate gate network. Figure 1(a) is a unate network but Figure 2(a) is not a unate network.

If a logic function is positive unate in x_i^*, then only x_i^* need be applied as an input to a restricted gate network realizing f; i.e., we need not apply $\overline{x}_i^*$ as an input.

We will be developing test sets for detecting multiple faults in restricted gate networks realizing positive unate functions and arbitrary logic functions. It is useful to develop a definition of input vectors, slightly different from the usual one, to these networks.

The truth table for a n-variable logic function, as defined normally, is a table of 2^n rows or 2^n n-dimensional binary input vectors and one column of binary functional values. It will be useful in the next section to define the input vectors over input literals (or a set of expanded input literals) instead of over input variables.

Notation: The truth table of a positive unate function u over the set of literals $\{x_1^*, x_2^*, x_3^*, \ldots, x_n^*\}$ is written as given in Figure 2(a). An example, the truth table of the function $\overline{a} + b\,\overline{c}$, is given in Figure 2(b).

The truth table of a logic function which is positive unate in the set of literals $\{x_{i_1}^*, x_{i_2}^*, \ldots, x_{i_\ell}^*\}$ and not unate in the set of variables $\{x_{j_1}, x_{j_2}, \ldots, x_{j_s}\}$ is written as that of a positive unate function over the expanded set of literals $\{x_{i_1}^*, x_{i_2}^*, \ldots, x_{i_\ell}^*, x_{j_1}, \overline{x}_{j_1}, x_{j_2}, \overline{x}_{j_2}, \ldots, x_{j_s}, x_{j_{s+1}}\}$.

The n-dimensional ((n+s)-dimensional binary vectors) which are the rows of the truth table (expanded truth table) are called input vectors (expanded input vectors).

As an example, in Figure 3, the expanded truth table of the function $a\ b + \bar{b}\ \bar{c}$ is given. Notice that even though the length of the input vectors is larger in the expanded truth table, the number of these vectors does not increase.

Definition 5: An input vector (expanded input vector) $\underline{X}$ covers the input vector (expanded input vector) $\underline{Y}$ if and only if $\underline{X}$ has ones every where $\underline{Y}$ has ones. Thus we write $\underline{X} \geq \underline{Y}$. If $\underline{X} \ngeq \underline{Y}$ and $\underline{Y} \ngeq \underline{X}$, then we say that $\underline{X}$ and $\underline{Y}$ are not comparable.

Definition 6: A minimal true vertex (minimal expanded true vertex) of a logic function is the input vector (expanded input vector) which does not cover any other true vertex except itself. A maximal false vertex (maximal expanded false vertex) of a logic function is the input vector (expanded input vector) which is not covered by any other false vertex except itself.

For example, the minimal expanded true vertices of the function given in Figure 3 are [0011, 1100] and the maximal expanded false vertices are [0101, 1010].

Definition 7: A set of test inputs T is called a complete set of tests for a logic function f if all stuck-at-faults in any restricted gate network realization of f can be detected by the application of T.

In the next section we will show that the set of minimal expanded true vertices and maximal expanded false vertices of a logic function f constitute a complete test set for f.

II. COMPLETE TEST SETS

In Theorem 1 we give an extension of Betancourt's[1] result which shows that for any restricted gate network realization of a positive unate function the set of maximal false vertices and minimal true vertices constitutes a sufficient test set to detect all s-a-0 and s-a-1 faults in the network. Theorem 1 is extended in Theorem 2 to include all logic functions.

In the sequel we will be using $U(\underline{X})$ or $F(\underline{X})$ to mean the output of the network under test, that is realizing the logic function u or f, respectively.

Theorem 1: In any restricted gate network realization of a positive unate function u, all stuck-at-faults can be detected by applying the maximal false vertices and minimal true vertices of u as test inputs.

Proof: Proof is obtained by showing that if $U(\underline{X}) = u(\underline{X})$ for every test input, (i.e., the output of the network under test, $U(\underline{X})$, equals the functional value $u(\underline{X})$) then $u(\underline{Y}) = u(\underline{Y})$ for every input. Assume u is positive unate over the literals $x_1^*, x_2^*, \ldots, x_n^*$. Let $U(\underline{X}) = u(\underline{X})$ for every test input and $u(\underline{Y}) = 1$ and $u(\underline{Z}) = 0$ for some $\underline{Y}$ and $\underline{Z}$. $u(\underline{Y}) = 1$ implies that $\underline{Y} \geq \underline{X}$, $\underline{X}$ is some minimal true vertex. If $\underline{Y} = \underline{X}$ then $U(\underline{Y}) = u(\underline{Y}) = 1$ by assumption and hence let $\underline{Y} \geq \underline{X}$ and $\underline{Y} \neq \underline{X}$. Therefore $\underline{X}$ can be changed to $\underline{Y}$ by changing some entries in $\underline{X}$ from zero to one and let these correspond to the positions $i_1, i_2, \ldots, i_\ell$. Clearly the outputs of all the gates in the network connected to $x_{i_1}^*, x_{i_2}^*, \ldots, x_{i_\ell}^*$ will change from 0 to 0 or 0 to 1 or 1 to 1 when $x_{i_j}^*$'s change from 0 to 1 and by repeating the argument we see that the output of the network changes from 0 to 0 or 0 to 1 or 1 to 1.[2] But $U(\underline{X}) = 1$ and hence the output of the network changes from 1 to 1 when the input changes from $\underline{X}$ to $\underline{Y}$ and hence $U(\underline{Y}) = 1$. By similar arguments we can show that $U(\underline{Z}) = 0$ for every $\underline{Z}$ such that $u(\underline{Z}) = 0$. Therefore if $U(\underline{X}) = u(\underline{X})$ for all test inputs then $U(\underline{X}) = u(\underline{X})$ for every input $\underline{X}$; which is the same as saying that the network correctly realizes u if for no test input $\underline{X}, U(\underline{X}) \neq u(\underline{X})$ and if for some test input $U(\underline{X}) \neq u(\underline{X})$ clearly the network under test is not realizing u.

Q.E.D.

Next we give a straightforward extension of Theorem 1 to arbitrary n-variable logic functions. Let f be unate in the set of input literals $X_u = \{x_{i_1}^*, x_{i_2}^*, \ldots, x_{i_\ell}^*\}$ and not unate in $X_{nu} = \{x_{j_1}, x_{j_2}, \ldots, x_{j's}\}$. By arguments very similar to the ones used in the proof of Theorem 1, we can prove Theorem 2 which is given next.

Theorem 2: The set of minimal expanded true vertices and maximal expanded false vertices of f are sufficient to test for all s-a-0 and s-a-1 faults in any restricted gate realization of f.

In the proof of Theorems 1 and 2 we have not assumed that the networks under test be irredundant and hence the test given are sufficient if there were redundancies in the network. This fact is especially important in the light of Friedman's[2] observation that faults in redundant leads may make some test sets invalid. We can further show that each one of the tests given in Theorem 2 is a necessary test for some two level realization of f in a manner similar to the one due to Betancourt.[1]

---

[2] Notice that this happens since the restricted gate networks contain AND and OR gates only and their outputs cannot change from 1 to 0 if the inputs change from 0 to 1.

The major drawback of the test sets of Theorem 2 is the fact that these can be very large. For logic functions which are not unate in any input variable, no two expanded input vectors are comparable and hence the complete test set contains all the 2^n input vectors. Of course if we want to test any realization of such logic functions, we cannot expect to do any better. But if we put some reasonable restrictions on the networks realizing the logic functions, then we may be able to reduce the size of the test sets.

Algorithms to design certain classes of restricted networks and to derive test sets to detect all stuck-at-faults in these networks have been developed and are reported in reference 3.

It is not difficult to see that Theorem 1 is valid for unate gate networks also (Definition 4). Results of Theorem 2 can be seen to hold good for those gate networks which can be changed to restricted gate networks (by applying DeMorgan's laws alone change NAND gates to NOT-OR and NOR gates to NOT-AND leaving the topology of the network intact). An example given next will illustrate the results in this section.

As an example, we will derive a complete test set for the function f_1, whose truth table is given in Figure 4(a) and the expanded truth table in Figure 4(b). The minimal expanded true vertices of f_1 are {01010, 01100, 10000} and the maximal expanded false vertices are {01111, 10101, 10110}. The input vectors (related to variables a, b, c and d) corresponding to these expanded vectors are {0010, 0100, 1000} and {0111, 1101, 1110} and these test inputs will detect all stuck-at-faults in an arbitrary restricted or unate gate network realizing f_1, even if there were redundancies in these networks.

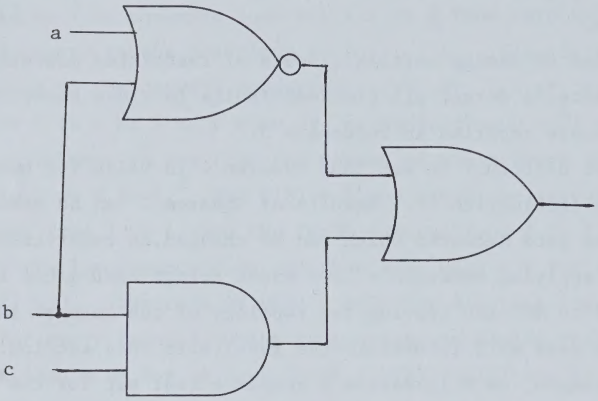
III. CONCLUSIONS

We have given an extension of an earlier result due to Betancourt,[1] to derive test sets for multiple stuck-at-faults for any realization of an arbitrary n-variable logic function. Furthermore, the test sets given will remain valid even in the presence of redundancy in the network being tested.

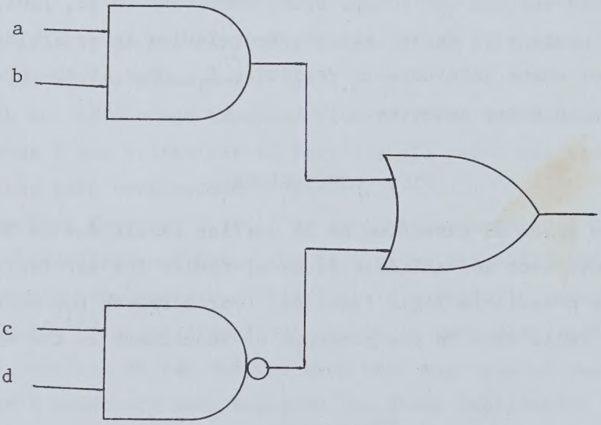
REFERENCES

1. Betancourt, R., "Derivation of Minimum Test Sets for Unate Logical Circuits," IEEE Trans on Computers, C-20, pp. 1264-1269, 1971.
2. Friedman, A.D., "Fault Detection in Redundant Circuits," IEEE Trans on Electronic Computers, EC-16(1), 99-100, 1967.

3. Reddy, S.M., "A Class of Testable Gate Network Realizations for Logic Functions," Technical Report No. 57, Office of Naval Research Project-The Theory and Applications of Automaton Theory, Division of Mathematics, University of Iowa, Iowa City, Iowa, May, 1972.



(b) Not a Unate Network



(a) A Unate Network

FIGURE 1: Examples of Gate Networks.

x_1^*	x_2^*	$\cdots$	x_n^*	u
0	0	. . .	0	0
0	0	. . .	1	0
⋮	⋮		⋮	⋮
1	1	. . .	1	1

(a)

$\bar{a}$	b	$\bar{c}$	u
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(b)

FIGURE 2: The Truth Tables of Positive Unate Functions.

a	b	$\bar{b}$	$\bar{c}$	f
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

FIGURE 3: Expanded Truth Table of $a\,b + a\,\bar{b}\,\bar{c}$.

a	b	c	d	f_1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

(a)

a	$\bar{a}$	b	c	d	f_1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1

(b)

FIGURE 4: Truth Tables of f_1.

SOME RESULTS ON TOTALLY SELF-CHECKING CHECKERS*

SUDHAKAR M. REDDY
Department of Electrical Engineering
University of Iowa
Iowa City, Iowa 52240

SUMMARY

A new formulation of the problem of designing totally self-checking checkers[4] for m-out-of-n codes is given. The following results are obtained.

1. Several checkers for K-out-of-2K codes. The important properties of these checkers are:
    a. two-level networks compared to at least four levels of logic required for checkers given earlier.[3-4] An example of a checker for 3-out-of-6 code is given in Figure 1;
    b. multi-level checkers requiring only 2K test inputs to detect stuck-at-faults. An example of a checker for 3-out-of-6 is given in Figure 2.
2. Checkers for K-out-of-(2K + 1) and (K + 1) out-of-(2K + 1) codes are given. There was no known systematic algorithm for this case until now. These checkers require only two-level networks.
3. Checkers for 1-out-of-3 and 1-out-of-7 codes. These were open problems until now.[4]

REFERENCES

1. Carter, W.C. and Schneider, P.R., "Design of Dynamically Checker Computers," IFIP 68, Vol. 2, Edinburg, Scotland, pp. 878-883, August 1968.
2. Carter, W.C., Jessep, D.C., and Wadia, A.D., "Error-Free Decoding for Failure-Tolerant Memories," Proc. 1970 IEEE International Computer Group Conf., Washington, D.C., pp. 229-239, June 1970.
3. Carter, W.C., Bouricius, W.G., Jessep, D.C., Roth, J.P., Schneider, P.R., and Wadia, A.B., "A Theory of Design of Fault-Tolerant Computers Using Standby Sparing," Digest 1971 International Symp. on Fault-Tolerant Computing, Pasadena, pp. 83-86, March 1971.

---

[2] The research reported has been supported in part by ONR grant N0001468A-0500.

4. Anderson, D.A., "Design of Self-Checking Digital Networks Using Coding Techniques," Coordinated Science Laboratory, Report R-527, University of Illinois, Urbana, Illinois, September 1971.
5. Reddy, S.M. and J.R. Wilson, "Testable Realizations for Some Logic Functions," submitted to IEEE Trans. on Computers.

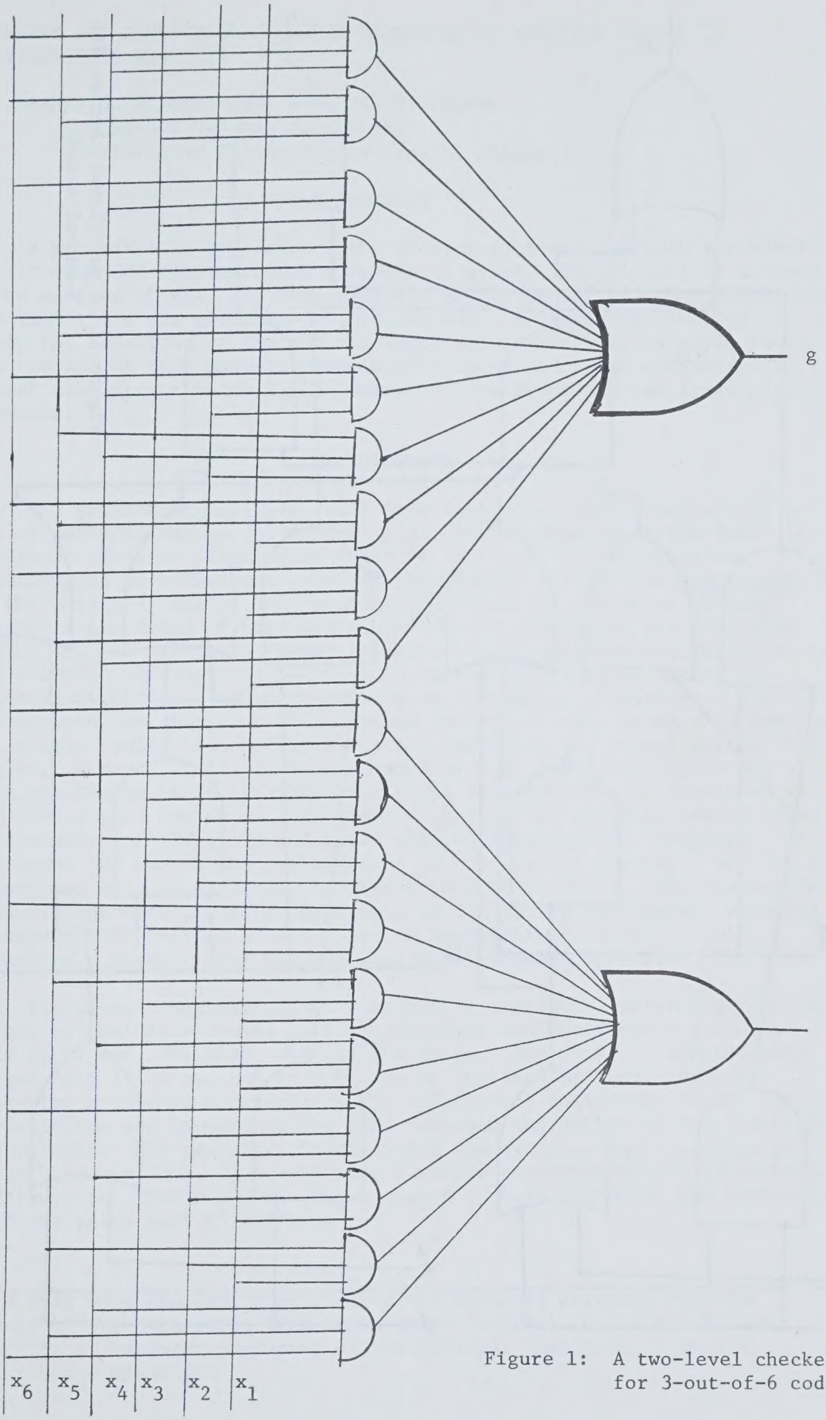

g
h
x_6
x_5
x_4
x_3
x_2
x_1

Figure 1: A two-level checker for 3-out-of-6 code.

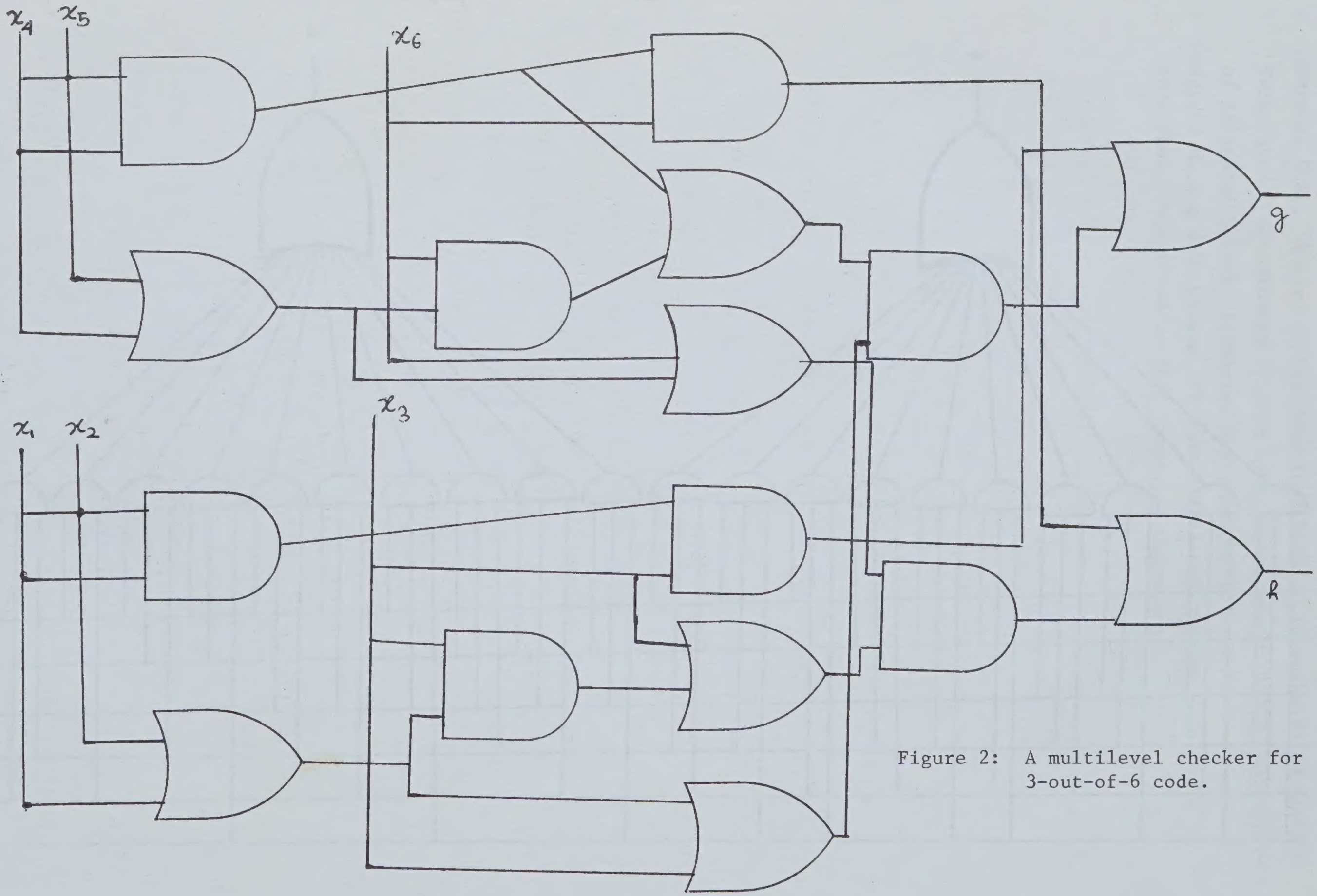

x_4
x_5
x_6
x_1
x_2
x_3
g
h

Figure 2: A multilevel checker for 3-out-of-6 code.

A THEORY AND A PROCEDURE FOR THE DETECTION OF MULTIPLE FAULTS IN COMBINATIONAL CIRCUITS *

SARMA R. VISHNUBHOTLA AND YING H. CHUANG
Computer Systems Laboratory
Washington University, St.Louis, Missouri

ABSTRACT

A procedure to construct a set of test patterns for both redundant and irredundant combinational circuits is given. The circuit is assumed to be made up of AND, OR, NAND, NOR and NOT gates and no other restrictions are imposed on the structure of the circuit. A theory is developed to study the behaviour of the circuit under the presence of multiple fault, and the set of test patterns constructed is proved to be sufficient to detect all detectable multiple faults in both redundant and irredundant circuits.

INTRODUCTION

The problem of multiple fault detection deals with constructing a set of patterns capable of detecting any (detectable) multiple fault in the given circuit. A multiple fault is caused by the simulteneous occurence of more than one stuck-at type faults, called the components of the multiple fault. This problem has drawn the attention of many people in the field of hardware diagnosis and some works are published [1,2,3,4], each giving a theory relating single and multiple fault dection and a method to construct patterns to detect all multiple faults in irredundant circuits with constraints on the circuit structure. Hayes [1] has shown by an example that a minimal set of patterns capable of decting all single faults (denoted by D) is not sufficient to detect all multiple faults. Schertz et al.[2] have shown that D is sufficient to detect all multiple faults in an irredundant circuit with two levels. Gault et al. [3] have proved that D is sufficient to detect all multiple faults in an irredundant multilevel tree network with alternating AND, OR gates. In the above two cases, the restrictions on the circuit structure are too severe and are not satisfied by many practical circuits. The cause-effect analysis approach presented by Bossen et al.[4] for irredundant circuits appears to be powerful as there are no restrictions on the circuit structure. However they did not consider redundant circuits.

The present approach gives a method to construct patterns which can detect all multiple faults in both redundant and irredundant circuits made up of AND, OR, NAND, NOR and NOT gates. Any fault is undetectable if and only if it can not be detected by any test pattern. Unless otherwise mentioned the word 'fault' refers to a detectable fault. No restrictions are imposed on the structure and the nature of the inputs of the circuit. The procedure to construct the patterns has three parts: 1) path enumeration, 2) sensitizing function construction and 3) pattern construction. Parts 1) and 3) are common to both redundant and irredundant circuits while part 2) differs.

---

* This work has been supported by the Advanced Research Projects Agency of the Department of Defence under Contract SD-302 and by the Division of Research resources of the National Institute of Health under Grant RR-00396.

PRELIMINARIES

An input variable appearing at two or more inputs either in the complemented or uncomplemented form is called a branching primary input variable. A variable which is not branching is nonbranching. A test pattern (or simply a pattern) is a combination of 1 and 0 on all primary inputs. A gate is a primary input (output) gate, if at least one of its inputs (outputs) is a primary input (output). The rest of the gates are called the internal gates. A path is said to be sensitized [5,6] if a signal can flow along it from input to output unimpeded. If G is a gate in a path, then all inputs of G which are not in the path are called the out-of-path inputs of G with respect to this path. If a value on any one input of G decides its output value, irrespective of the values on the other inputs of G, then this input value of G is called the dominating input value of G and its complement is called the sensitizing value of G. A wire serving as input to two or more gates is called a branch-out wire and each of its branching wire segment is called a branch wire. Whenever there is a branch-out wire, a path including this wire branches into several segments. All segments starting with the first branch-out wire are called co-branches. The (inversion) parity of a co-branch is 0 or 1 depending on whether the number of inversions in the co-branch is even or odd. Primary inputs not in a path are called its off-path inputs.

PATH ENUMERATION

In this section a procedure to enumerate the paths is given. Paths are classified into three groups: 1) simple paths, 2) multiple paths and 3) subpaths. When none of the co-branches belonging to a branch-out wire reconverge, we include all these co-branches in a path and call it a simple path. When some of the co-branches reconverge, all reconverging co-branches with one parity and all nonreconverging co-branches are included in one path. The reconverging co-branches with the other parity (when they are more than one) are all included in a different path. We denote the former path by P and the latter by P'. P or P' is called a multiple path if it includes at least two reconverging co-branches. Whenever there are reconverging co-branches, each of these co-branches together with the segment before the first branch-out wire and that after the reconvergence is considered as a path called a subpath. The path enumeration rules are as follows.

1) Let G be a primary input gate with nonbranching primary inputs. Starting with all such inputs of G, trace the wires forward. If there is no reconvergence of co-branches then this will be a simple path. Otherwise enumerate a path (say P) including all reconverging co-branches of one parity and another (say P') including all reconverging co-branches of the other parity. If there are reconverging co-branches, then enumerate subpaths each including one of these co-branches.

2) Let x be a branching primary input variable. Consider all input wire segments starting with x or x' (complement of x) as co-branches, and enumerate the paths as in rule 1).

Example 1. Consider the circuit in figure 1. Different numbers are given to a branch-out wire and its branches even if all of them are logically equivalent. This distinction is necessary because there can be a fault on a branch wire while the corresponding branch-out wire is fault free.

By applying the rules of path enumeration we obtain the following paths. $P_1 = (1,6,15,20,24)$; $P_2 = (1,8,21,24)$; $P_3 = (2,5,20,24)$; $P_4 = (3,14,23,24)$; $P_5 = (4,11,22,24)$; $P_6 = (4,13,19,23,24)$. All these six paths are subpaths as they include only one reconverging co-branch each. Paths $P_7 = (2,9,12,16,17,18,19,21,22,23,24)$ and $P_8 = (3,7,10,15,16,17,18,20,21,22,24)$ are multiple paths each including three reconverging co-branches of the same parity. Three subpaths are enumerated for each of these multiple paths, obtaining six following subpaths. $P_9 = (2,9,16,17,21,24)$; $P_{10} = (2,9,16,18,22,24)$; $P_{11} = (2,12,19,23,24)$; $P_{12} = (3,10,16,17,21,24)$; $P_{13} = (3,10,16,18,22,24)$; $P_{14} = (3,14,23,24)$.

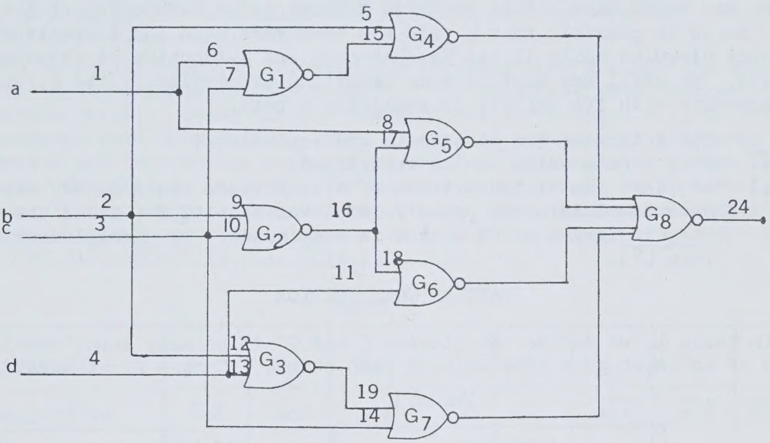


Figure 1

SENSITIZING FUNCTION

The concept of path sensitization [5,6,8] has been widely used in fault diagnosis. As the paths enumerated fall into three types, we shall discuss the rules for sensitizing these different types. A simple path can be sensitized by fixing all of its out-of-path inputs at their sensitizing values. For multiple paths and subpaths, fix all out-of-path inputs of the path at sensitizing values, while unsensitizing all reconverging co-branches not in the path by fixing one of their out-of-path input at a dominating value. The wire conditions 0 and 1 of a wire w are expressed by w' and w respectively.

Fixing the out-of-path inputs at sensitizing or dominating values can not be accomplished directly as the internal wires in the circuit are not directly accessible. So there is a need for a systematic method to map the various internal wire conditions into the primary input values, and the concept of a sensitizing function (S-function) [6] is introduced for this reason.

1) Take the AND combination of all sensitizing out-of-path wire conditions of the path and call it the sensitizing AND combination. If there are some reconverging co-branches which are not included in the path (which should therefore be unsensitized), then take the OR combination of all dominating wire conditions of the out-of-path inputs of a co-branch and call it unsensitizing OR combination and repeat this for all such

co-branches. Take the AND combination of all these OR combinations and call it the unsensitizing AND combination, and finally take the AND combination of the sensitizing and unsensitizing AND combinations and call it the resultant AND combination.

2) Express the resultant AND combination as sum of products of the primary inputs. Each product term is called an S-term.

3) Choose an S-term with the fewest literals. If the circuit is irredundant, then this S-term is the S-function of the path.

4) If the circuit is redundant, then take the AND combination of maximum number of S-terms such that their AND combination is not zero and call it an M-term. Repeat this for the remaining S-terms until all the S-terms are taken care. Each of these M-terms is an S-function of the path. So it is possible that a path may have more than one S-function in redundant circuits while it can have at most one S-function in irredundant circuits. We shall now mention some important properties of the S-function in connection with its ability to sensitize a path.

Lemma 1. The following two statements are equivalent.

a) P is a path which can be sensitized.

b) At least one of the S-terms of P is nonzero and none of these S-terms contain any primary input variable of P (called the on-path inputs of P) either in complemented or uncomplemented form [9].

PATTERN CONSTRUCTION

In table 1, we define two classes C and C' of primary input combinations of an input gate from which a path starts. Choose a path and fix all of its off-path inputs at values specified by the S-function chosen. Combine all the combinations in C and C' of the path with the off-path input combination to obtain t+1 patterns. Notice that all these patterns will have the same values on their off-path inputs. If the circuit is redundant, repeat this for all S-functions. Apply this for all paths enumerated and call the set of patterns obtained as D_k.

Table 1

	AND/NAND	OR/NOR
	1 2 3 ---------- (t-1) t	1 2 3 ---------- (t-1) t
C	1 1 1 ---------- 1 1	0 0 0 ---------- 0 0
C'	0 1 1 ---------- 1 1	1 0 0 ---------- 0 0
	1 0 1 ---------- 1 1	0 1 0 ---------- 0 0
	1 1 0 ---------- 1 1	0 0 1 ---------- 0 0
	1 1 1 ---------- 0 1	0 0 0 ---------- 1 0
	1 1 1 ---------- 1 0	0 0 0 ---------- 0 1

Theorem 1. The set of patterns D_k is sufficient to detect all single faults in both redundant and irredundant circuits [9].

Lemma 2. If P is a path which can be sensitized, then the set of patterns constructed for P detects all faults in P under single fault assumption (with the exception of those faults which can not be detected by sensitizing P) [9].

Bounds on the number of patterns: Suppose there are N paths in the circuit and t_i inputs to the i-th path. Then in the case of irredundant circuits the D_k will have at most $\sum_{i=1}^{N}(t_i+1)$ patterns. In redundant circuits D_k will have at most $\sum_{i=1}^{N} m_i(t_i+1)$ patterns, m_i being the number of M-terms for the i-th path. We shall now illustrate S-function and pattern construction for the circuit of figure 1.

Example 1. (continued): Consider the path $P_1 = (1,6,15,20,24)$. The branch (8,21), which is a reconverging co-branch of (6,15,20) but which is not included in P because of the difference in parity, should be unsensitized while sensitizing P. So we obtain

$$\begin{aligned}
&\text{Sensitizing AND combination} &&= \overline{7}.\overline{5}.\overline{21}.\overline{22}.\overline{23}\\
&\text{Unsensitizing OR combination} &&= (17 + 22 + 23)\\
&\text{Resultant AND combination} &&= (17 + 22 + 23)\ \overline{5}.\overline{7}.\overline{21}.\overline{22}.\overline{23}
\end{aligned}$$

$$= \overline{5}.\overline{7}.17.\overline{21}.\overline{22}.\overline{23} = \overline{b}.\overline{c}.16\ (8 + 16)(11 + 16)(14 + \overline{12}.\overline{13})$$

$$= \overline{b}.\overline{c}.\overline{12}.\overline{13} \qquad = \overline{b}.\overline{c}.\overline{d}$$

So we obtain only one S-term $\overline{b}.\overline{c}.\overline{d}$ and this does not contain the primary input variable of this path "a". So we choose this term as the S-function for P_1. Using the two combinations a = 0 and a = 1, we obtain two patterns abcd = 0000,1000. This process is repeated on all paths enumerated and the various S-functions and the patterns constructed from them are tabulated in table 2. The six subpaths from P_9 to P_{14} are ignored as none of these subpaths can be sensitized. From the remaining eight paths we obtain D_k consisting of the following ten patterns. abcd = 0000, 0001,0010,0100,1000,1111,0111,1011,1101,1110.

Table 2

Paths	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
S-functions	$\overline{b}\overline{c}\overline{d}$	bcd	acd	abd	abc	$\overline{a}\overline{b}\overline{c}$	$\overline{a}\overline{c}\overline{d}$	$\overline{a}\overline{b}\overline{d}$
Patterns	0000 1000	0111 1111	1011 1111	1101 1111	1110 1111	0000 0001	0000 0100	0000 0010

MULTIPLE FAULT DETECTION IN IRREDUNDANT COMBINATIONAL CIRCUITS

Assuming a multiple fault f_k, we analyze the effect of its fault components on signal flow and show that D_k is capable of detecting all multiple faults. If all branch wires of a branch-out wire are stuck at the same value, then we replace all these faults by a single fault stuck-at the same value on the branch-out wire because its effect is the same as the faults on the branch wires. Let P be a path which has some components of f_k such that the set Q of patterns constructed for P detects at least one of the components under single fault assumption. These components lying in P are called the on-path components of P and the rest, the off-path components of P. As P can be either a simple path, a multiple path or a subpath, we shall show that D_k detects f_k in all the three cases. In each case the study of the effect of f_k is divided into two parts. In the first part we ignore the presence of off-path components and study the effect of on-path components on signal flow. In the second part we take the effect of off-path components into account. Whenever there are reconverging co-branches, there will be a gate beyond which none of these co-branches reconverge. Such a gate is called a final gate of reconvergence. In multiple output circuits there may be more than one such gate.

Suppose f_1 is a fault component in P which can be detected by Q. From the wire of f_1 onwards, signal flow along any branch of P is decided by f_1 because all segments in P are sensitized by every pattern in Q. Suppose there is another component f_2 in a branch along which f_1 flows. Then f_2 decides the flow of the faulty value along the branch from the wire of f_2 onwards, and we say that f_2 masks f_1 along this branch. If two components f_1 and f_2 produce faulty signal flows along two branches which eventually converge such that they give a sensitizing value and a dominating value respectively at the inputs of a gate, then we say that f_2 dominates f_1 at this gate. The on-path component which can not be masked or dominated by any other on-path component is called a maximal on-path component of P.

Consider only the off-path components which can affect the signal flow in the path P. The off-path component which can not be dominated by any other off-path component with respect to any pattern in Q is called a maximal off-path component of P.

Consider a maximal on-path and a maximal off-path component of P such that the faulty values produced by them converge into a gate in P. If one of these components dominates over the other with respect to all patterns in Q, then the dominating component is called a maximal effective component. If neither of these components dominate over the other, then both of them are maximal effective components. Recall the way the two classes C and C' of primary input combinations of P are used to obtain Q. Under no fault condition the patterns obtained from C and C' give complementary values on all output wires of P. If a single fault is detectable by sensitizing P, then it is detected by a pattern obtained from C or C'. This pattern together with any pattern obtained from the other class (with the same off-path primary input values of course) are called the complementary patterns for the fault. Henceforth whenever we say the complementary patterns of P, we mean the pair of complementary patterns for the fault which is detectable by sensitizing P.

Theorem 2. Let P be a simple path in the circuit with at least one maximal effective component in the path, then D_k detects f_k.

Proof: P being a simple path, none of its co-branches reconverge if they exist. If the maximal effective fault is a maximal on-path component, then it will fix the value of at least one of the output wires of P at the same value with respect to both complementary patterns of P. This implies that one of the two patterns detects f_k. On the other hand if the maximal effective fault is an off-path component, then it remains to be a maximal effective fault with respect to either of the complementary patterns of P. This is because the segments along which the maximal effective fault enters into P are sensitized by both complementary patterns, as their sensitization always depends only on the off-path primary input values for simple paths and these are the same for all patterns in Q. So the maximal effective fault fixes the value of one output wire at the same value with respect to both complementary patterns and hence f_k is detected by one of these patterns, i.e. D_k detects f_k. Q.E.D.

Theorem 3. Let there be a multiple path or subpath P in the circuit. If there is a maximal effective fault on P or forcing a faulty signal into P, beyond the final gate of reconvergence G of all the reconverging co-branches starting at the branch-out wire of P, then D_k detects f_k.

Proof: If the maximal effective component is an on-path component, as it is beyond G it fixes at least one output wire of P at a value identical for both complementary patterns of P. So one of them detects f_k. If the maximal effective component is an off-path component, then as the convergence of the faulty signal into P takes place beyond G, the sensitization of the segment along which the faulty signal flows does not depend on the primary input values of P. That is, the sensitization of this segment depends only on the off-path primary input values of P which are the same for both complementary patterns. So this fault component again fixes one of the output wires of P at the same value with respect to both complementary patterns and so one of them detects f_k. Hence D_k detects f_k in either case. Q.E.D.

Theorem 4. Let there be a multiple path or subpath P in the circuit. If there is a maximal effective component in P or forcing a faulty value into P not beyond the final gate of reconvergence G, then D_k detects f_k.

Proof: In a multiple path there may be a maximal on-path component on more than one reconverging co-branches. But none of these maximal on-path components can dominate over the others, for otherwise the dominated one can not be a maximal on-path component. So if the maximal on-path component in P is not beyond G, it will also force a faulty value across G and show up on the output of P. Hence if the maximal effective component is a maximal on-path component, then one of the complementary patterns of P for this fault component will detect f_k. If the maximal effective component is a maximal off-path component, then (by definition) it will not be dominated by any other off-path components. Also it will not be dominated by any maximal on-path component. So the complementary patterns of P for this maximal off-path component will detect f_k. This proof holds for subpaths also as a subpath is a special case of a multiple path with only one reconverging co-branch included. Hence in either case D_k detects f_k. Q.E.D.

Theorem 5. If f_k is a detectable multiple fault, then there will be at least one path in the circuit with a maximal effective component.

Proof: Consider the fault components and the segments starting from each of these components and ending in a circuit output. Either if none of these segments can be sensitized or if every faulty value flow due to a component is masked by an undetectable fault, then f_k will be an undetectable fault. As f_k is detectable, there will always be a component at least one of whose faulty signal flow is not masked by any undetectable component of f_k along a segment that can be sensitized. So there will be a path containing this as a maximal on-path component, which is either a maximal effective component by itself or dominated by a maximal off-path component which will therefore be a maximal effective component.

Q.E.D.

Theorem 6. If f_k is any detectable multiple fault in the circuit, then D_k detects f_k.

Proof: As f_k is a detectable multiple fault, by theorem 5, there will be at least one path in the circuit containing a maximal effective component. This path will have to satisfy the hypothesis of one of the theorems 2,3 or 4 and in either case D_k detects f_k. Q.E.D.

MULTIPLE FAULT DETECTION IN REDUNDANT COMBINATIONAL CIRCUITS

Friedman [7] has observed that a fault detectable as a single fault may become undetectable under the simultaneous presence of another redundant single fault in redundant circuits. He called the former fault a second generation redundant fault and the latter, a first generation redundant fault. When there is a multiple fault f_k in the circuit, the definitions of first and second generation redundant faults have to be generalized. The set of all components of f_k which are on the redundant wires is called the set of first generation redundant faults and is denoted by F. The set of all components of f_k not in F, and which becomes undetectable because of F, is called the set of second generation redundant faults and is deonted by S. The number of components in F, S and F U S (where U is the operation of set union) are denoted by $|F|$, $|S|$, $|F \cup S|$ respectively. Let $|F \cup S| = q$. If all components in f_k are either in F or S, then f_k will be undetectable. So if f_k is detectable, then q is less than k.

Lemma 3. Let D_k be the set of patterns obtained for a redundant circuit. Any component f, of f_k which is detectable as a single fault will become a second generation redundant fault, if and only if, f is masked or dominated by the components in F with respect to every pattern in D_k.

Proof: Suppose f is detectable as a single fault. If f becomes a second generation redundant fault, then evidently by the definition of a second generation redundant fault, f is masked or dominated by one or more components in F with respect to every pattern in D_k.

Conversely, suppose f is masked or dominated by the components in F with respect to every pattern in D_k. The M-terms constructed for any path contain the sensitizing information contained in all S-terms of this path. So the set of all M-terms constructed for a path sensitizes every sensitizable segment in the path at least once. Now consider the segments starting from f and ending in the circuit outputs. The fact that f is masked or dominated by the components in F with respect to every pattern in D_k implies that f is masked or dominated by the components in F, whenever any segment starting with f and ending in a circuit output is sensitized. So there is no other way for the faulty signal of f to reach a circuit output under the simultaneous presence of F in the circuit. This shows that f is a second generation redundant fault. Q.E.D.

Theorem 7. In any redundant combinational circuit made up of AND, OR, NAND,NOR and NOT gates, if f_k is a detectable multiple fault, then the set of patterns D_k obtained for this redundant circuit detects f_k.

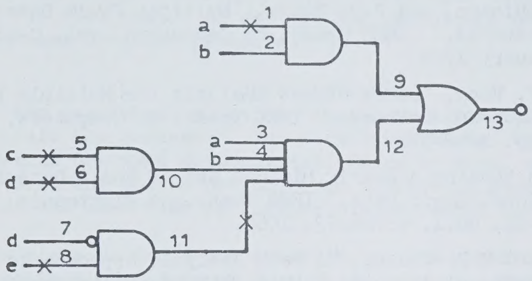
Proof: As f_k is a detectable multiple fault, it will have at least one component f_1 not in F U S. By Lemma 3, there will be a pattern in D_k detecting f_1 as a single fault such that no fault in F U S can either mask or dominate f_1. This is true for all components of f_k not in F U S. Ignoring all the components of f_k which are not in F U S, we get another multiple fault $f_{(k-q)}$. Theorems 2,3,4 and 5 are applicable for redundant circuits also when f_k is replaced by $f_{(k-q)}$, because the first and second generation redundant components of f_k which could have possibly affected their applicability for this circuit have all been removed without loss of detectability of f_k by D_k. So we see that D_k detects $f_{(k-q)}$, i.e. D_k detects f_k. Q.E.D.

Example 2. The circuit of figure 2, is an extension of the one given by Friedman [7] to show the presence of second generation faults. Wires 5,6, 7,8,10 and 11 are all redundant. The only paths that can be sensitized are $P_1 = (1,9,13)$ and $P_2 = (2,9,13)$. To sensitize P_1, wires 2 and 12 have to be fixed at values 1 and 0 respectively. Also any of the wires 4, 10 or 11 have to be fixed at 0. So we obtain

$$2 \cdot \overline{12}(\overline{6} + \overline{10} + \overline{11}) = b(\overline{3} + \overline{6} + \overline{10} + \overline{11})(\overline{6} + \overline{10} + \overline{11})$$
$$= b(\overline{6} + \overline{10} + \overline{11}) = b(\overline{b} + c + \overline{d} + d + e)$$
$$= b\bar{c} + b\bar{d} + bd + be.$$

The four terms $b\bar{c}$, $b\bar{d}$, bd and be are the S-terms for P_1.

Ignore the term $\bar{a}b$ as it contains the variable "a" which is the primary input variable of P_1. The four terms $b\bar{c}$, $b\bar{d}$, bd and $b\bar{e}$ are the S-terms of P_1. If the circuit were an irredundant circuit, we could have chosen any one of these S-terms as the S-function. But because this is a redundant circuit, ignoring some S-terms might hinder the detectability of some detectable faults. For example consider the faults f_1, f_2, f_3, f_4 and f_5. Under single fault assumption f_4 is a detectable fault as it can be detected by the pattern abcde = 11XXX (where X denotes a don't care). But under the simultaneous presence of f_1, f_2, f_3 and f_5 the fault f_4 can not be detected by any pattern and so f_1, f_2, f_3 and f_5 which are the first generation redundant faults make f_4 a redundant fault which is therefore a second generation redundant fault.



Faults:
f_1 = wire 5 s-a-1
f_2 = wire 6 s-a-1
f_3 = wire 8 s-a-1
f_4 = wire 1 s-a-0
f_5 = wire 11 s-a-1

Figure 2

If P_1 is sensitized by choosing any one of the S-terms $b\bar{c}$, $b\bar{d}$, $b\bar{e}$ as the S-function, then the patterns constructed may not detect the multiple fault caused by f_1, f_2, f_3 and f_4. For example,choose $b\bar{c}$ as S-function and obtain the two patterns abcde = 11000 and 01000. None of these patterns can detect this multiple fault. However if we choose bd as S-function, then the pattern abcde = 11011 detects this multiple fault. However if we choose only bd as the S-function, then the patterns constructed from it, abcde = 11011 and 01011, can not detect another multiple fault with f_1, f_4 and f_5 as its components, whereas it can be detected by choosing $\bar{b}d$ as the S-function. This shows that to detect all detectable multiple faults in redundant circuits, the sensitization information included in all the S-terms must be taken into account and so the concept of M-terms is necessary.

From the four S-terms of P_1 we obtain the two M-terms, $M_1 = \bar{b}\ \bar{c}\ \bar{d}\ \bar{e}$ and M_2 = bd. Similarly for P_2 we obtain two M-terms $M_1 = a\ c\ \bar{d}\ e$ and M_2 = ad. So the D_k for this circuit consists of the six patterns abcde = 11000, 01000, 11X1X, 01X1X, 10000 and 10X1X. By theorem 7, these six patterns detect any detectable multiple fault in this circuit.

REFERENCES

(1) J.P. Hayes, A Study of Digital Network Structure and its Relation to Fault Diagnosis, Ph.D. dissertation, University of Illinois, Urbana, 1970.

(2) D.R. Schartz and G.A. Metze, "On the Indistinguishability of Faults in Digital Systems," Proc. 6th Annual Allerton Conf. on Circuit and System Theory, pp. 752-760, October 68.

(3) J.W. Gault, J.P. Robinson, and S.M. Reddy, "Multiple Fault Detection in Combinational Networks," IEEE Trans. on Computers, vol. C-21, no.1, pp.31-36, January 1972.

(4) D.C. Bossen and S.J. Hong, "Cause-effect Analysis for Multiple Fault Detection in Combinational Networks," IEEE Trans. on Computers, vol. C-20, no.11, pp. 1252-57.

(5) D.B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets," IEEE Trans. on Electronic Computers, vol. EC-15, no.1, pp.66-73, 1966.

(6) S.R. Vishnubhotla and Y.H. Chuang, "A Path Analysis Approach to the Diagnosis of Combinational Circuits," Proc. 8th Design Automation Conference, June 1971, pp. 222-230.

(7) A.D. Friedman, "Fault Detection in Redundant Circuits," IEEE Trans. on Electronic Computers, February 1967, pp. 99-100.

(8) H.Y. Chang, E.G. Manning and G. Metze, Fault Diagnosis of Digital Systems, Wiley-Interscience, 1970.

(9) S.R. Vishnubhotla and Y.H. Chuang, "A Path Analysis Method for the Diagnosis of Multiple Faults in Combinational Circuits," Technical Memorandum No. 155, April 1972, Computer Systems Laboratory, Washington University, St.Louis, Missouri.

NETWORKS AND VEHICLE ROUTING FOR MUNICIPAL WASTE COLLECTION

EDWARD J. BELTRAMI AND LAWRENCE D. BODIN
Urban and Policy Sciences
State University of New York
Stony Brook, New York 11790

ABSTRACT

Vehicle routing for municipal waste collection encompasses a variety of problems. In this paper, we explore the techniques we have developed for solving some of these problems.

INTRODUCTION

One of the persistent questions facing many municipalities is "what do we do with all the garbage?" In operational terms, this means that communities must decide how many disposal facilities to have, where to locate these facilities, and how to get the ever increasing amounts of refuse to them while at the same time satisfying environmental and economic constraints.

In this paper we focus our attention on some operational problems regarding waste collection. Most of what we discuss is a direct outgrowth of our experience as members of the Technical Advisory Group to the New York City Environmental Protection Agency (which includes the Department of Sanitation).

Collection activities of the New York City Department of Sanitation include household and bulk collection by sanitation crews working on trucks and barges as well as street cleaning and snow removal. About 80% of the $200 million annual operating budget for the New York City Department of Sanitation is related to wages and benefits for the 11,000 sanitationmen and most of this is in connection with refuse collection. However, little work has been carried out so far on the optimal utilization of this workforce.

In this paper, we consider one aspect of this question; namely the efficient routing of collection vehicles. Other facets of this problem will not be treated here. These include questions of how to optimally deploy the workforce (this area has been discussed elsewhere by Bodin (1) and Altman et.al (2)), the effect of innovations in collection technology,and the optimal location of disposal facilities. It is however, apparent that all of these questions are interrelated. For example, if the disposal facilities are inadequately designed or if there are too few of them available, then delays occur because of the congestion of vehicles waiting to dump. Furthermore, such delays can affect the routing of the vehicles by reducing the amount of time a truck has for collection.

The magnitude of the collection and disposal task in New York City becomes further apparent from the fact that each day about 25,000 tons of solid waste need to be collected and disposed of and about 11,000 miles of streets must be cleaned. Thus any attempt to render these services more efficient is likely to result in substantial dollar savings. The studies described in this paper have produced noticeable cost benefits to New York City.

In what follows we describe a set of procedures for routing vehicles through a set of nodes in a network or along its branches. Although the task is one of collection with disposal as a terminal operation at the dumpsites, the problem is not too unlike that of the truck delivery problem. In that case, a dump is replaced by a depot and the pickup points become delivery points.

For these reasons, the procedures in this paper are of general interest.

An expanded version of the results described herein will appear in a forthcoming issue of the Journal of Networks.

ROUTING THROUGH NODES

Most refuse collection activities in New York City center around the pickup of household refuse in small bins. However, large institutional sites such as schools, hospitals, and apartment complexes have their refuse stored in large 6-12 cubic yard metal containers (equivalent to 40-80 normal garbage cans). Eventually about 1000 such sites will need to be serviced by using large capacity trucks which depend on mechanical fork lifts to load the bins into the chassis where it is then compacted (These are called hoist compactor trucks). Each truck can service several such sites before going to a dump to unload.

The problem we considered was how to route the hoist compactor trucks to minimize the total travel time of the vehicles and to determine the minimum number of trucks needed each day. This last condition is important from the point of view of minimizing the capital expenditure needed to outfit a fleet of trucks.

There is also a dual problem to the one mentioned above. That is, for a given number of trucks one seeks the maximum number of locations that can be serviced. We have found several organizations who have either purchased or leased a fleet of trucks interested in this question since they want to know how far they can expand their operations without additional capital expenditure.

The problem is not simply one of finding the shortest path through a network since there are a number of complicating factors. Hence, it becomes quickly apparent that the usual graph theoretic arguments are not directly applicable. First, one must be mindful of capacity and time constraints. Each pickup point can have a different quantity to be picked up and, since the capacity of the truck is limited, the route must be interrupted for travel between pickup points to dumps. Moreover, there are several dump sites. Having saturated the truck, the problem asked is which dump site should be used? Finally, some locations require daily service while others do not. Since there are typically many points to service, the problem is not only to arrange the routes feasibly but to assign each pickup point to days of the week to minimize the number of trucks needed.

An analysis is based on a modification of the heuristic Clarke and Wright idea (ref. 14). The basic Clarke and Wright method, quite simply, is this. Assume, to begin with, that a tour consists of visiting one containerized location and then returning to the depot or a dumpsite. If now two locations are joined together to form one tour then the "savings in total travel time" is illustrated in figure 1. The savings S_{ij} between pickup points i and j is $S_{ij} = T_{oi} + T_{oj} - T_{ij}$ where T_{ij} is the time to go between the point i to j and T_{oi} (or T_{oj}) is the time to go from dumpsite to pickup point i (point j). One then calculates the "savings" associated with all pairs of locations to be serviced and then sorts in decreasing order the list of pairs with positive savings. Starting at the top of the list, points are combined provided that the resulting tour is feasible from the point of view of time and capacity. In this way increasingly longer and better tours are formed until the list is exhausted.

Several comments are in order. First the ability to generate tours by this approach does not preclude the possibility that some tours can be improved (total time to cover tour reduced) by altering the order in which the points on the tour are visited. This is done by solving the traveling salesman problem. Rather than developing a general traveling salesman code, we used the 2-opt and 3-opt heuristics of Lin (6) to approximate the optimal solution to the traveling salesman problem.

Second, the simple generation of tours does not completely solve the problem. It happens that some sites need to be serviced 3 times a week while other sites require 6 times a week service (there is no garbage pickup on Sunday except in circumstances which we can ignore here). The question that arises is which 3 time a week sites shall be serviced on Monday, Wednesday, Friday and which 3 time a week sites shall be handled on Tuesday, Thursday, Saturday. There are several possible ways of solving this problem two of which are described below.

One approach (first described in (17) is to form tours and then to assign these tours to either Monday, Wednesday, and Friday or Tuesday, Thursday, and Saturday. The second approach is to preassign sites to specific days of the week and develop routes based on this assignment. We have implemented both procedures in the study of hoist compactor routing in New York City.

In the first approach, we used the Clarke and Wright procedure to generate the tours but made all 6 times a week nodes appear twice in the network. However, in forming tours, such a site and its image were not allowed to appear on the same tour. This forced the 6 time a week locations to appear on two different tours, each of which is assigned to different days of the week. If there are k six time a week sites in the original network of n vertices then the network with its images consists of n+k sites. In this algorithm, one is sometimes able to join site i+n (the image of site i) with site j even when i cannot be joined to j (because of the subtours that sites i and j are on).

Having completed the first part of this algorithm, one has a collection of tours which are so far unassigned to either Monday, Wednesday and Friday or Tuesday, Thursday, and Saturday. The problem now arises as to which days of the week to assign these tours to. This can pose a difficulty. It is not obvious that from a given collection of tours a feasible day assignment can be found.

We were able to successfully handle this feasibility question by building into the computer code a routine which prevented the cycle as indicated in the above example to occur. As there werefew six day a week collection points a good solution occurred. If there were many six day a week collection points the second appraoch imbedded within a simulation would be recommended because it would have been very difficult to find a nearly optimal feasible solution using the first approach. The simulation randomly assigns sites to either Monday, Wednesday, Friday or Tuesday, Thursday and Saturday and solves two Clarke and Wright routing problems; one on the points assigned to Monday, Wednesday, Friday, and the second on the points assigned to Tuesday, Thursday, Saturday. The random assignment of sites to days is repeated several times. The best Clarke and Wright solution from these random assignments is then taken as the optimal set of routes.

It is interesting to note that after experimenting with the first appraoch quite extensively, the second approach was actually used for routing the hoist compactor trucks for the New York City Department of Sanitation. This was done because the Department of Sanitation decided a priori the day assignment for each site.

It should be mentioned in passing that since the methods used are heuristic it may happen that imposing additional constraints can sometimes yield better results. We found this to be true in one case in which certain tours were blocked to prevent the pathology associated with multiple service frequencies. This somewhat counterintuitive result can be explained by the fact that heuristic algorithms can sometimes lock on to local solutions which are not nearly globally optimal. The use of constraints may have the effect of smoothing out the constraint region and, hence, forcing the tours to be ordered in a different and perhaps better way.

An additional refinement has to do with multiple dumps. The basic Clarke and Wright algorithm assumes that each truck disposes of its load at a single location. If there are several such disposal locations we pre-assign each collection site to the dump closest to it. The "savings" of combining two points assigned to the same disposal location is computed as before. However, the savings achieved by combining points assigned to two different dumpsites is now $S_{ij} = T_{oi} + T_{1j} - T_{ij}$, where T_{oi} (or T_{1j}) is the time to go between dumpsite o and site i (or between dump 1 and site j) and T_{ij} is travel time between sites i and j (see figure 2).

A final question to be considered in the routing of the hoist compactor trucks is that of forming day schedules for each of the trucks. It is a typical occurrence that these tours do not exceed the time constraints imposed by union regulations and, hence, several such tours can be hooked together to form a day schedule for the truck. Generally, the capacity of the hoist compactor truck is the binding constraint in generating routes by the algorithms of the Clarke and Wright type. Since one of our objectives is to minimize the total number of trucks needed, some ways of forming day schedules are better than others. We again used heuristic techniques for concatonating together routes to form day schedules.

RANDOMIZED TOURS

A second routing question we encountered was the problem of routing and dispatching of barges and tugboats for the disposal of 10,000 tons of refuse daily at a landfill on Staten Island. The refuse is at 8 marine disposal locations around New York City (brought there by trucks, as discussed above) and is then hauled away by barges. There are two shifts a day with a resulting daily average flow of 16 barges back and forth between the landfill and each of the marine disposal points.

Many of the operational questions here are similar to the ones discussed above but there are some additional complications due to the fact that one has to account for tides (tugs move slowly, if at all, against the tide) and because each tug can haul as little as one barge per trip. Since it is not necessary to pick up a full complement of loads at each site it visits, a given transfer station can be visited on more than one trip each day.

One of the basic restrictions in forming truck tours as described in the previous section is that each node is to be visited at most once on a given day. However, there may be some advantage in visiting some points several times in one day, by different vehicles on different routes. Figure 3a exhibits the only tour possible when each truck has maximum capacity of 3 (and a maximum travel time of 6), if we insist on visiting each node only once. The total travel time is 12. However, in Figure 3b we reduce the travel time to 10 and use only two tours by the simple expedient of allowing site 2 to be serviced twice, where each time half of its daily requirement is taken care of. In general, this kind of split service at certain key nodes can be expected to cut down the number of tours required to satisfy demands, especially when these nodes are such that their requirements are nearly the same as the capacity of the vehicles, which is the case in barge routing.

What we suggest is that it may be advantageous to generate tours by combining the Clarke and Wright algorithm with a randomized search procedure. The way this is done is explained in the expanded version of this paper to appear in the Journal of Networks.

BRANCH ROUTING

A third class of problems which we were asked to investigate by the New York City Department of Sanitation was the routing of the street sweepers or, as they are sometimes called, the mechanical brooms. These vehicles, which are trucks having rotating brooms attached to their front, sweep the streets of a road network by moving along these streets in a continous path. This continuous path is a sequence of streets that have to be covered and streets that are traversed but need not be swept (called deadheaded streets).

For a given area of New York City, we know, because of the City's alternate side of the street parking regulations, which side of the street has to be swept for each time period. Furthermore, each time period is separated from the other time periods by at least two hours; therefore, we can solve the routing problems for each time period separately. For each time period and a street network representing a given area, there is a subset of streets that have to be covered. The other streets in the original street networks are used for deadheading the vehicles between the streets that are to be covered.

The starting and ending times of the shift for the driver are known and there are no restraints on how much refuse the brooms can pick up. The brooms are to be at the starting point of the tour when the tour is scheduled to commence, and the brooms are to return to the garage at the end of the day. However, we did not have to account for these times in the routing of the vehicle because the hours that the alternate side of the street parking regulations are in effect are well contained within the starting and ending times of the shift for the driver.

The problem we have, therefore, is to service the street network that has to be covered in each time period so that the total amount of deadheading by all the brooms assigned to the area is minimized. This deadheading time can occur from either a vehicle traversing a street that has to be covered more than the required number of times or travelling at least once over a street that does not have to be swept. By reducing the deadheading as much as possible, we can either cut down on the number of brooms that are needed to service a given area or provide a better service to this area with the brooms we now have available. Better service in this case means traversing a street at a slower speed or going over a street more than once during a time period.

Because the sweepers have to travel in the direction of traffic and because of the alternate side of the street parking regulations, we consider all the branches in the network to be directed. Thus, for a two-way street, this allows us to distinguish between the side that is to be covered and the side which can be used for deadheading. There are other classes of branch routing problems whose networks are either undirected or a mixture of undirected and directed. Two examples of this in the public domain are the routing of snow plows and the routing of trucks for household refuse collection.

As cited earlier, variants of this problem have been of interest to mathematicians and operations researchers for many years. One version of this problem was mentioned by Euler in 1736 in a celebrated paper in which he showed that when all the nodes of the network are of even degree and all the branches in the network are undirected, then a tour through all the branches without deadheading in the network can be found (Ref. 22). A tour through all the branches in the network, regardless of whether the network has directed branches, undirected branches, or both is called an Euler tour (Edmonds and Johnson (11)).

The basic premise in all the procedures designed for solving branch routing problems with one vehicle is to add branches to the subgraph containing only the branches which have to be covered to guarantee that when all the branches are added a Euler tour exists in the network. The branches added to the original network are selected to insure that the sum of the distances of all the branches added to the network is minimized (Edmonds and Johnson (11), Liebman (9), Stricker (8), and Liebling (10)). There are two variants of the single vehicle routing procedures for solving the mechanical broom routing problem involving more than one vehicle.

The first procedure finds a giant tour and busts this tour into a set of routes which are feasible with regard to the time capacity of the vehicle. The second method breaks the geographic area over which the routes are to be formed into a collection of contiguous regions, the number of such regions equal to the number of brooms to be routed, and then solves the branch routing problem over each of these subregions. Once again, details of our approach will appear in the subsequent article of the Journal of Networks.

ACKNOWLEDGEMENT

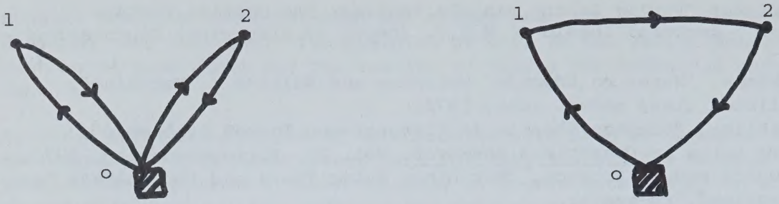
A number of our colleagues at Stony Brook have contributed substantially to portions of this work, notably Stanley Altman and Naresh Bhagat. Their involvement has been as essential ingredient to the success of the Stony Brook routing studies.

Research was in part supported under the N.S.F. Grant GI-5

References

1. L. Bodin, "Towards a General Theory of Manpower Scheduling - Parts 1 and 2", to appear in the Journal of Urban Analysis, 1972
2. S. Altman, E. Beltrami, S. Rappaport, and G. Schoepfle, "A Nonlinear Programming Model for Household Refuse Collection", IEEE Systems, Man and Cybernetics, Vol. SMC-1, July 1971, 289-291.
3. M.J. Krone, "Heuristic Programming Applied to Scheduling Problems", unpublished doctoral dissertation, Princeton University, 1970.
4. D. Marks, J. Cohon, H. Moore, and R. Stricker, "Routing for Municipal Services", Proceedings of the 6th Annual ACM Urban Symposium, 1971.
5. M. Held and R. Karp, "The Travelling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, No 6, pp 1138-1162.
6. S. Lin, "Computer Solutions of the Travelling Salesman Problem", Bell System Technical Journal 44 (1965), pp.2245-2269.
7. J. Little, K. Murty, D. Sweeney, and C. Karel, "An Algorithm for the Travelling Salesman Problem", Operations Research, Vol, 11, No. 6. (1963), pp. 972-989.

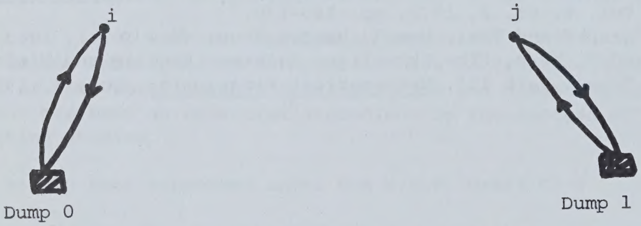
8. R. Stricker, "Public Sector Vehicle Routing: The Chinese Postman Problem", doctoral thesis at M.I.T. (Dept. of Electrical Engineering) 1970.
9. J. Liebman, "Notes on Edmonds' Matching and White's K. Matching", unpublished class notes, June, 1972.
10. T. Liebling, "Graphen Theorie in Planungs-und Touren problemen", Lecture notes in Operations Research, Vol. 21, Springer-verlag, 1970.
11. J. Edmonds and E. Johnson, "Matching, Euler Tours and the Chinese Postman Problem", to appear.
12. N. Christofides, "The Optimum Traversal of a Graph", Report 71/16 of Imperial College of Science (London), November 1971.
13. R. Newton and W. Thomas, "Developing a Computer Program for Bus Routing: Final Report", State University of New York Research Foundation, Albany, New York, July 1970.
14. G. Clarke and J.W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research 12 (1964), pp, 568-581.
15. S. Eilon, Watson-Gandy, and N. Christofides, "Distribution Management", Hafner Publishing Co. (New York) 1971.
16. B. Bennett and D. Gazis, "School Bus Routing by Computer", to appear in Transportation Research.
17. S. Altman, N. Bhagat and L. Bodin, "Extension of the Clarke and Wright Algorithm for Routing Garbage Trucks", presented at the TIMS 8th International Meeting (Washington, D.C.) March 1971.
18. S. Benzer, "On the Topology of the Genetic Fine Structure", Proc. Nat. Academy of Science USA 45 (1959), pp. 1607-1620.
19. D.G. Kendall, "Incidence Matrices, Interval Graphs and Seriation in Archeology", Pacific J. Math 28 (1969), pp.565-570.
20. A. Tucker, "Perfect Graphs and an Application to Optimizing Municipal Services", submitted for publication.
21. P. Krolak, W. Felts, and J. Nelson, "A Man-Machine Approach Toward Solving the Generalized Truck-Dispatching Problem", Transportation Science, Vol. 6, No. 2, 1972, pp. 149-170.
22. O. Ore, "Graphs and Their Uses", Random House (New York), 1963.
23. M. Held and R. Karp, "The Travelling Salesman Problem and Minimum Spanning Trees: Part II", Mathematical Programming, Vol, 1, 1971 pp. 6-25.



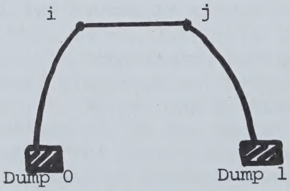
Initial Routing Procedure:
Total travel time is twice travel from dump to site 1 plus twice travel between dump and 2.

Modified Routing Procedure:
Total travel time is time between dump and site 1 plus time between dump and 2 minus time between sites 1 and 2.

Figure 1



Initial routing in case of two dumps.



Routing procedure which results in a savings

Figure 2

Figure 3a

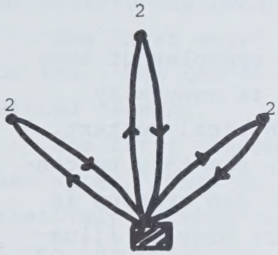
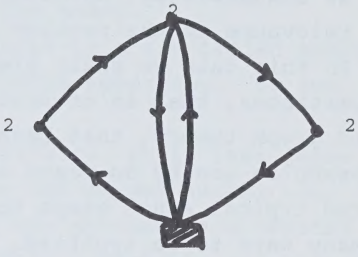


Figure 3b



The numbers indicate the requirements or loads at each of the nodes. Vehicle capacity is 3.

GRAPH THEORY PROBLEMS IN PHYSICS

DANIEL J. KLEITMAN
Massachusetts Institute of Technology
Cambridge, Massachusetts

1. INTRODUCTION

Graphs form a means of representing binary relations. As such, they may be introduced with virtually any context in any field of knowledge. In many circumstances the aspects of graphs that are often studied in "graph theory" such as connectivity, coloring problems, and enumeration, bear relevance to the problem at hand.

In this talk we shall present several examples of such circumstances, that is of problems in what is generally called graph theory, that have arisen in physical context. The examples are by no means exhaustive nor can they be considered typical since graph theory problems can crop up in too many ways to be typified. At best, they provide illustrations of what can happen.

Our first examples concern discrete models in statistical mechanics. The final example will concern enumeration of perturbations theoretic diagrams.

2. DISCRETE MODELS IN STATISTICAL MECHANICS

We begin by reminding the reader of the nature of statistical mechanics. From our point of view it is a remarkably simple subject.

Our model of reality, whatever it is, is characterized by a space (or possibly a set) S of configurations, on the points of which a certain "energy" function $H(p_i,x)$ is defined (the p_i here represent certain parameters).

The main object of statistical mechanics is computation of (or computation of properties of) the "partition function" $Z(p_i,T)$ which is the sum or integral over all configurations of $e^{-H/kT}$.

From the partition function many properties of the system can be deduced. Thus by taking appropriate derivatives with respect to T and parameters p_i, specific heats and susceptibilities of the system may be obtained. The entropy of the system is related to the logarithm of the partition function (with the lowest energy value set equal to zero). Among the most interesting aspects of the partition function are its analyticity properties as a function of T. A real positive point of singularity in T indicates the existence of a phase transition in the model.

Now let us turn to describe some models. One standard approach to the study of physical phenomenon procedes by attempting to construct the simplest possible model that exhibits the behavior under investigation. One can then attempt to show that the added complications of reality do not destroy the behavior seen in the model.

The first model we will consider is a simplified model for ice. Now ice is a very remarkable substance, for two related reasons.

The first reason is that it floats. It is less dense than water. That is, the hydrogen and oxygen nuclei are relatively further separated from one another in a crystal of ice than they are in water. (This remarkable and almost unique property has had great influence on the evolution of life).

The second reason is that it violates the so called "third law of thermodynamics". The latter says that the entropy of any subject approaches zero as T approaches 0. Interpreting the entropy as the logarithm of the partition function, at zero temperature this merely becomes the logarithm of the number of lowest energy configurations of the system. For most substances there is a unique or almost unique lowest energy state and those substances obey the third law. For ice, the number of lowest energy configurations seems to approach C^N where N is the number of atoms, so that the entropy becomes a constant (proportional to log C) per particle.

Our model for ice is planar. We imagine the oxygen nuclei located at integer coordinate points in a plane, with the hydrogen nuclei located along lines joining adjacent oxygens. Since hydrogens like to be closer to their neighboring oxygens (as they are in water) then they would be if half way between them, they can lie on their line at either of two points closer to either oxygen end of their line. The energy H in our model is zero when the hydrogens lie at these points whenever each oxygen has two hydrogen's near it, and two far away. (To avoid edge effects we might identify opposite edges of our lattice).

A configuration here consists of a location for each hydrogen in one of its two positions. The problem of computing the entropy per oxygen of ice at zero temperature becomes the problem of enumerating zero energy configurations of hydrogens.

The problem can be stated in a number of alternate ways. First, we can denote the statement that a hydrogen is nearer an oxygen by affixing an arrow on the hydrogen line pointing at that oxygen's end of it. The relevant problem then becomes:

Given an mxm (say) square grid of lines, in how many ways can one-way arrows be affixed to the lines so that at every interior point two arrows point inward and two outward?

This is already a problem in graph theory. We are given a graph and asked in how many ways can it be made into a digraph respecting the two in two out restriction at each vertex?

By some fooling around one can state the problem in the following way:

How many ways can one color the mxm grid in two colors such that every interior point lies on exactly one boundary between colors (i.e., no 4 boxes around one point are the same color, and no checkerboard like pattern occurs at any point)?

I am not aware of any direct especially graph-theoretic solutions of this problem, although one can obtain bounds on the solution by straightforward efforts. Lieb was able to solve this problem by means geared more to the rectangular nature of the array than to its structure as a graph. He found that the number of solutions behavior as

$(\frac{4}{3})^{(3m^2/2)}$ in the limit of large m, a result not far from the observed entropy results for ice at lowest temperature.

Of course ice is not planar, and this model is not therefore realistic. Results have been obtained on other models of this kind. Moreover by defining a suitable energy function H in this system the model can be used to exhibit behavior of ice at non zero temperature, and also to model ferroelectricity.

Our second model can be described in very similar terms. It is the so called two dimensional Ising model of a ferromagnet, the first model shown (by Onsager) to exhibit a phase transition. One has a grid as above but now a configuration consists of arrows placed upon the (integer) points of the grid pointing normal to the plane-either up or down.

(These arrows represent in this model the tiny magnets whose behavior produce the phenomena known as ferromagnetism. The observed behavior is that in such substances some such tiny magnets seem to all line up together at temperatures below the transition temperature, while they tend to have no long range correlations at higher temperatures).

Here the energy function H depends merely on the number of up arrows and the number of neighboring pairs of arrows which have opposite orientation.

A graph theoretic problem related to the problem of computing the partition function here is in how many ways can one color the mxm grid in two colors given that x squares are of one color and the boundary consists of y lines.

It is interesting that the partition function here exhibits a transition point, a point of non-analyticity in the limit of large m.

A third problem with some of the same flavor is that of finding the lowest energy state configurations for certain alloys. Suppose a substance has (say) a face centered cubic lattice structure and suppose it has a given proportion of "alloy" atoms which occupy sites in the lattice.

If the energy H is given as $\lambda, N_1+\lambda_2 N_2$ where N_1 is the number of nearest neighbor pairs of alloy atoms and λ_2 the number of second nearest neighbor pairs, what configurations minimize H, as a function of λ_1, λ_2 and p? There is one case, I believe in which the minimum energy value such in configurations has not been proven to be so.

3. GRAPHS IN PERTURBATION THEORY

There is a quite different area of physics in which graphs as diagram's are often employed. In describing interacting physical particles or systems it is common to expand the function under consideration in a "perturbation series" in powers of the interactory term(s) in the energy function H. The terms in such an expansion correspond to possible arrangements of various powers of the interaction terms and are often described by diagram's or graphs. The desired description of the physical system is often expressed as the sum over all possible diagrams of some function characterized by the diagram.

Occasionally one is led to enumerate diagrams having certain properties; and then the problem often becomes an enumeration problem in graph theory - how many graphs can be constructed having such and such properties?

Here is an example of a problem which arose in this manner. Imagine a circle upon which 2n points are distinguished. These may be paired off by straight lines required in (2n-1)(2n-3)..., ways. We say that a pairing is reducible (or k-reducible) if it is possible to cut the circle into two parts, each containing some of our points by a straight line which crosses none (or less than k) of the pairing lines. We can ask what proportion of the possible graphs are reducible (or k-reducible)?

For this particular problem the solution turns out to be that for large n a proportion $\frac{1}{e}$ of the pairings are not reducible, and a proportion $e^{-(2k-1)}$ are not k-reducible, which results can be obtained by combinatorial techniques.

The examples above do not even touch many important areas. For example enumerations of chemical isomers are a traditional subject for application of the Burnside-Polya graph counting techniques. Study of cluster expansions, and of diffusion through study of random walks involve enumeration problems in graph theory. Rather than futilely attempting to be exhaustive I will stop here.

A selected annotated bibliography is included.

BIBLIOGRAPHY

L. Onsager, Phys. Rev. 65, 117 (1944).

The exact solution of the rectangular Ising model n partition function is given. Both high and low temperatures expansions, related by a dual transformation, are connected with combinatorial problems in graph theory. Generalization to triangular and hexagonal lattices is briefly discussed by introducing a star-triangle transformation.

J. Ashkin and E. Teller, Phys. Rev. 64, 178 (1943).

Introduce Ashkin-Teller model, generalization of Ising model. The dual transformation is discussed using graph theory.

R. B. Potts, Proc. Camb. Phil. Soc. 48, 106 (1952)

L. Mittag and M. J. Stephen, J. Math. Phys. 12, 441 (1971)

Higher component two-dimensional Ising models (Potts models) are introduced and their transition points are found by Potts. Mittag and Stephen examine graph-theoretic representations with respect to dual transformations.

G. F. Newell and E. Montroll, Rev. Mod. Phys. 25, 353 (1953)

Early review article giving clear exposition of graphical problems related to the Ising model.

P. Kasteleyn, J. Math. Phys. 4, 308 (1963)

Discovers connection between Ising models and dimer graphs, closely related to Pfaffians. Whole new approach begins, culminating in solution to the domino problem: How many ways can 32 dominoes cover a chess board?

E. H. Lieb, Phys. Rev. 162 162 (1967)

Zero point entropy of two-dimensional ice is calculated exactly. The ice condition is essentially a statement about directed graphs.

A. Lenard, unpublished comment

The ice condition is equivalent to placing squares of three different colors on a grid such that no two adjacent squares are similarly colored.

R. J. Baxter, Phys. Rev. Lett. 26 832 (1971)
Exact solution of eight-vertex model which includes many previous models: ice; ferroelectric; KDP, Ising etc., is given. The problem is stated originally in graph-theoretic language.

E. H. Lieb, F.Y. Wu, Two Dimensional Ferroelectric Models, Chap. II. The equivalence between many physical models, related to the eight-vertex model, and graph-theoretic problems is demonstrated. (This may be published as a Handbook Article).

L. Onsager, J. F. Nagle, M-S Chen, and J. C. Bonner, Hopping of ions in ice,(preprint). The migrations of ions in real ice is (three-dimensional lattice) represented by a full Cayley tree enabling a calculation of the density of states for a simple Hopping Hamiltonian.

D.J. Kleitman, Proportions of Irreducible Diagrams, Studies in Applied Mathematics, Vol. XLIX No. 3, pp. 297-299, 1970.

The alloy problem was considered in a master's thesis by S. Allen in the Metallurgy Department at Massachusetts Institute of Technology last year.

GRAPH THEORY IN DIAGNOSTIC PROBLEMS

WATARU MAYEDA
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

One way to improve diagnosability of a system is to insert additional components, which may just be terminals other than prespecified input and output terminals at which the tests for diagnosis can be performed. Depending on the kind of additional components we use, the condition for diagnosability may change. Here, we will discuss the diagnosability under three kinds of additional components, namely test points, test gates and test S-gates.

INTRODUCTION

When the diagnosability depends only on the topology of a system, then the employment of linear graphs yields to relate properties of systems and special subgraphs of corresponding linear graphs.

This relationship allows us to use linear graph theory to develop topological consitions for diagnosis of systems.

To improve diagnosability of a system, additional components will be inserted. The optimum locations and the optimum number of such additional components depends not only on the topology of a system but also on the kind of additional components to be employed. Here we will show how the locations and the number of additional components change by the use of different kinds of components. Especially, we will see the difference made by the use of test points, test gates and test S-gates as additional components.

LINEAR GRAPH FOR A SYSTEM

Here we will consider only those systems which will be represented by linear graphs [1] in such a way that each vertex represents a component and each edge between two vertices indicates the connection between two components represented by the vertices. Furthermore, we consider the case where only components represented by vertices can be defected. That is, connections represented by edges can never be defected. This assumption will not limit systems to be covered by this paper. The reason is that when there is a possibility that a connection between components will be defected, we can modify a linear graph by using a pair of series edges to represent the connection so that the defect of the connection will be indicated by the defect of the common vertex.

A fault vertex is a vertex which represents a defected component [1,2,3]. For convenience, we consider only the case that a signal which passes a fault vertex will be distorted and will remain to be distorted by passing another fault vertex. Hence, when we monitor a signal, we can determine whether the signal passes fault vertices or not.

DIAGNOSABILITY

Consider that a signal T_{ij} is injected at a vertex i and monitored at vertex j. From the structure of a linear graph G corresponding to a given system, we know which vertices that signal T_{ij} will pass through. The collection of all these vertices is called a measurable set Ω with respect to T_{ij}. By the result of monitoring T_{ij}, if we found that T_{ij} passes at least one fault vertex, then Ω contains the vertex. On the other hand, if monitoring T_{ij} indicates the absence of fault vertices we know that there is a fault vertex in linear graph G, then $\overline{\Omega} = \Omega_o - \Omega$ contains the fault vertex where Ω_o is the set of all vertices in G.

Suppose we made n tests by injecting $T_{i_pj_p}$ for $p = 1,2,\ldots,n$. Let Ω_p be the measurable set with respect to $T_{i_pj_p}$ [1,2]. Also suppose we know that there is a fault vertex in linear graph G. If monitoring $T_{i_1j_1}, T_{i_2j_2}, \ldots,$and $T_{i_uj_u}$ indicates the existence of a fault vertex and monitoring $T_{i_{u+1}j_{u+1}}, \ldots,$and $T_{i_nj_n}$ indicates the absence of a fault vertex, then

$$\Omega_1 \cap \Omega_2 \cap \ldots \cap \Omega_u \cap \overline{\Omega}_{u+1} \cap \ldots \cap \overline{\Omega}_n$$

contains the fault vertex.

For given measurable sets $\Omega_1, \Omega_2, \ldots, \Omega_n$, the collection $\{D\}$

$$\{D\} = \{D\ ;\ D = \tilde{\Omega}_1 \cap \tilde{\Omega}_2 \cap \ldots \cap \tilde{\Omega}_n\}$$

where $\tilde{\Omega}_p$ is either Ω_p or $\overline{\Omega}_p$ is called the D-partition under a set of test signals $T_{i_1j_1}, T_{i_2j_2}, \ldots, T_{i_nj_n}$. Suppose D in $\{D\}$ contains vertices v_1 and v_2. Then when either v_1 or v_2 is a fault vertex, we cannot distinguish which one is the fault vertex by the tests corresponding to measurable sets $\Omega_1, \Omega_2, \ldots,$and Ω_n. Hence we can define k distinguishability by the sets in $\{D\}$ as follows: A linear graph (or a system corresponding to a linear graph) is k-distinguishable under a set of tests if there is a set D in $\{D\}$ consists of k vertices and no sets in $\{D\}$ consist of more than k vertices. For example, if every set in $\{D\}$ consists of one vertex, a linear graph is 1-distinguishable. This means that when any one vertex becomes a fault vertex, we can find it by the prespecified tests by which the D-partition is obtained.

DIAGNOSABILITY WITH TEST POINTS

When a linear graph G has several entry vertices for test signals to be injected and several exit vertices for test signals to be monitored, then we can obtain the D-partition under the tests using these vertices. Suppose we found that G is k-distinguishable under these tests. If we wish to improve the value k, we must employ some other testing procedure. One way is to use vertices other than the entry and exit vertices for injecting and monitoring test signals. To see the effect by using these additional vertices, we assume that a given linear graph has only one entry vertex and only one exit vertex. Such a linear graph is known as an SEC (Single Entry Single Exit Connected) graph [1,2]. We can see that the distinguishability of SEC graphs can be extended to that of linear graphs with several entry and exit vertices.

Since there is only one entry vertex and only one exit vertex in a SEC graph, a test signal, which is injected at the entry vertex and monitored at the exit vertex, must pass all vertices in a SEC graph G. Hence the measurable set Ω_o with respect to this test contains all vertices in G. In other words, this test only indicates whether there is a fault vertex or not.

One way to test an SEC graph for locating a fault vertex is by inserting so called test points. To discuss test points, let's use the symbols $v(e^+)$ and $v(e^-)$ for indicating vertices where edge e is connected. The orientation of edge e is from vertex $v(e^+)$ to vertex $v(e^-)$. When a test point is inserted at an edge e, we can inject a test signal at vertex $v(e^-)$ and monitor a test signal at vertex $v(e^+)$ as shown in Fig. 1. Physically, this is equivalent to having an induction coupling at a connection represented by edge e. Hence, we can have two measurable sets one of which is with respect to the test by injecting a test signal at the entry vertex and monitoring it at vertex $v(e^+)$ and the other is with respect to the test by injecting a test signal at vertex $v(e^-)$ and monitoring it at the exit vertex.

When we insert test points at edges $e_1, e_2, \ldots,$ and e_n in a SEC graph with entry vertex i and exit vertex j, we can inject a test signal at any of vertices i, $v(e_1^+), v(e_2^+), \ldots,$ and $v(e_n^+)$ and monitor it at any of vertices j, $v(e_1^-), v(e_2^-), \ldots,$ and $v(e_n^-)$. Hence we have $(n+1)^2$ possible measurable sets including Ω_o. However, the D-partition obtained by using all these measurable sets and that obtained by using only 2n measurable sets of types $\Omega_{iv(e_k^+)}$ and $\Omega_{v(e_k^-)}$ $(k = 1, 2, \ldots, n)$ are the same. This means that there are at most 2n useful testing for diagnosis when n test points are inserted. Note that $\Omega_{iv(e_k^+)}$ is a measurable set associated with injecting a test signal at i and monitoring it at vertex $v(e_k^+)$ and $\Omega_{v(e_k^-)j}$ is a measurable set associated with injecting a test signal at vertex $v(e_k^-)$ and monitoring it at vertex j.

Let's use the symbol $\mathcal{E}(\Omega \times \overline{\Omega})$ as a set of all edges each of which is connected from a vertex in Ω to a vertex in $\overline{\Omega}$. When a test point is inserted at an edge e in an SEC graph with entry vertex i and exit vertex j, we can show [1] that $\mathcal{E}(\Omega_{iv(e^+)} \times \overline{\Omega_{iv(e^+)}})$ is a directed cut-set from i to j. Also $\mathcal{E}(\overline{\Omega_{v(e^-)j}} \times \Omega_{v(e^-)j})$ is a directed cut-set from i to j. We say that edge e covers these directed cut-sets. Let E be a set of edges at which test points are inserted. Since for each edge in E, there are directed cut-sets covered by the edge, we have a collection $\{S\}$ of directed cut-sets covered by edges in E. With this collection, we can state the following:

Theorem 1: If there is an edge in a SEC graph G which is not in any set in $\{S\}$, G is not 1-distinguishable.

DISTINGUISHABILITY WITH TEST GATES

A test gate [1,3] is one which can be inserted at an edge so that we can equivalently remove the edge whenever necessary. When we employ only test gates for diagnosis of a SEC graph with entry vertex i and exit vertex j, the location of injecting a test signal is always i and that of monitoring the test signal is always j. Thus every measurable set contains both i and j. This means that if either i or j is a fault vertex we cannot distinguish which one is a fault vertex. Hence we can never achieve 1-distinguishability. However if we disregard vertex j, we may obtain 1-distinguishability. Hence we define k-(ij) distinguishability which is k-distinguishability when we consider vertices i and j as one. In other words, if a SEC graph is 1-(ij) distinguishable, then we can pinpoint the location of a fault vertex except when either i or j is a fault vertex, we can only indicate that either i or j is a fault vertex.

When test gates are inserted at edges $e_1, e_2, \ldots, e_n$, then we will have different tests depending on which of $e_1, e_2, \ldots, e_n$ are removed prior to injecting a test signal at the entry vertex and monitoring it at the exit vertex.

Suppose we remove edges e_1, e_2, and e_3 and inject a test signal at entry vertex i and monitor it at exit vertex j. Then the measurable set contains all vertices which the test signal will pass under the absence of edges e_1, e_2, and e_3. This measurable set is symbolized by $\Omega_{(e_1 e_2 e_3)}$.

For employment of only test gates, we have the following property:

Theorem 2: Let E be a set of edges at which test gates are inserted. Also let a SEC graph G with entry vertex i and exit vertex j have no edges connected directly from i to j. Then G is not 1-(ij) distinguishable if E contains no semicut from i to j.

If $E = \{ab\}$ in a SEC graph in Fig. 2 with i and j being the entry and the exit vertices, then by the above theorem it is not 1-(ij) distinguishable. Note that $\{ab\}$ contains no semicut from i to j.

Since set E must contain a semicut from i to j for 1-(ij) distinguishability, let's consider the case where E itself is a directed cut-set from i to j. Let $E = \{e_1, e_2, \ldots, e_n\}$. There will be $2^n - 1$ measurable sets obtained by removing edges in E. However, we can show that the D-partition obtained by using all these $2^n - 1$ measurable sets and that obtained by using n measurable sets of type $\Omega_{(\overline{e}_k)}$ for $k = 1, 2, \ldots, n$ is the same when $\Omega_{(\overline{e}_k)}$ is the short hand notation of $\Omega_{(e_1 e_2 \ldots e_{k-1} e_{k+1} \ldots e_n)}$.

DISTINGUISHABILITY WITH TEST S-GATE

A test S-gate [1] can be inserted to a vertex so that we can equivalently remove the vertex whenever necessary. Physically this is equivalent to shorting the output terminal of a component corresponding to the vertex. Similar to the case when test gates are employed, we are considering k-(ij) distinguishability of a SEC graph under the use of test S-gates. Suppose test S-gates are inserted at vertices $v_1, v_2, \ldots, v_n$. Corresponding to removing some of these vertices, then injecting a test signal at entry vertex i and monitoring it at exit vertex j, we will have a measurable set consisting of all vertices which the test signal will pass through under this circumstance.

Let's define a vertex semicut as follows: A vertex semicut from i to j is a minimal set of vertices such that by deleting all vertices in the set, all directed paths from i to j will be destroyed. With this definition, we have the following interesting property:

Theorem 3: Let V be a set of vertices at which test S-gates are inserted. Then an SEC graph with entry vertex i and exit vertex j is not 1-(ij) distinguishable if V contains no vertex semicuts from i to j.

EXAMPLE AND FURTHER PROBLEMS

Consider an SEC graph with entry vertex i and exit vertex j as shown in Fig. 3. If we insert test points at edges a, b and c, then it is 1-distinguishable. If we insert test gates at edges a, b, and c, then it is 2-(ij) distinguishable. Similarly if we insert test S-gates at vertices $v(a^+)$, $v(b^+)$ and $v(c^+)$, it is 2-(ij) distinguishable.

For employing these three kinds of additional components, there is a common necessary and sufficient condition for 1 [1-(ij)] distinguishability which is as follows: For any two vertices (disregard exit vertex j for 1-(ij) distinguishability) if there is a measurable set which contains one of these vertices, then it is 1 [1-(ij)] distinguishable.

Since this condition is not easy to test, it is desirable to find necessary and sufficient conditions for k [k-(ij)] distinguishability which are hopefully topological. Also no distinguishability conditions have been found when employing not just one kind but two or more kinds of additional components together.

REFERENCES

1. Mayeda, W., Graph Theory, John Wiley & Sons, New York, 1972.

2. Mayeda, W. and Ramamoorthy, C. V., "Distinguishability Criteria in Oriented Graphs and its Application to Computer Diagnosis - I," IEEE Trans. on Circuit Theory, CT-6, 448-454, Nov. 1968.

3. Ramamoorthy, C. V. and Mayeda, W., "Computer Diagnosis Using the Blocking Gate Approach," IEEE Trans. on Computers, C-20, 1294-1299, Nov. 1971.

4. Manning, E., "On Computer Self-Diagnosis (Part I and Part II)," IEEE Trans. on Electronic Comput., EC-5, 873-890, Dec. 1966.

5. Preparata, F. P., Metze, G., and Chien, R. T., "On the Connection Assignment Problem of Diagnosable Systems," IEEE Trans. on Electronic Comput., EC-16, 848-854, Dec. 1967.

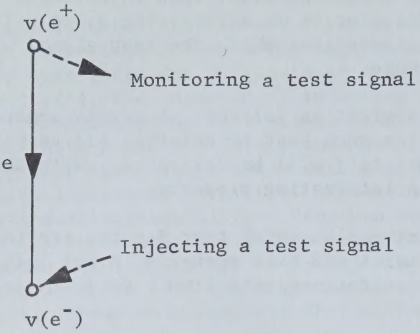

$v(e^+)$
Monitoring a test signal
e
Injecting a test signal
$v(e^-)$

Fig. 1. Insertion of a test point at edge e.

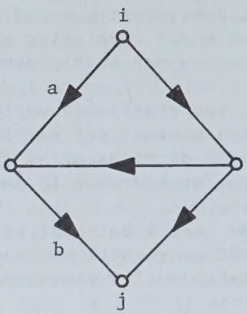


Fig. 2. SEC graph.

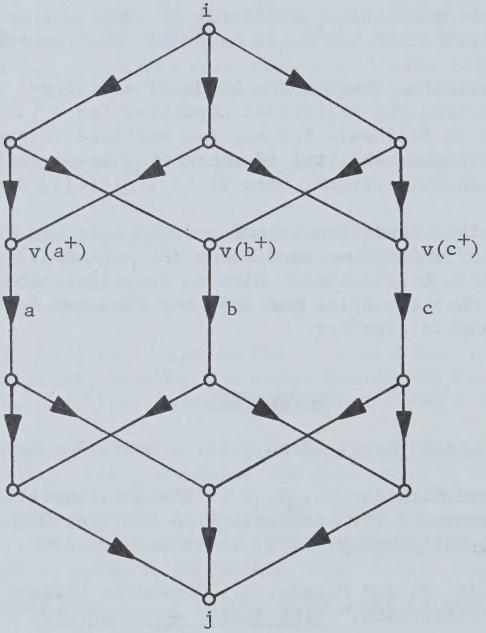

i
$v(a^+)$
$v(b^+)$
$v(c^+)$
a
b
c
j

Fig. 3. SEC graph.

RELIABILITY OF NETWORKS

R. VAN SLYKE
Network Analysis Corporation
Glen Cove, New York

ABSTRACT

Three promising approaches to the problem of relating the reliability of a network as a whole system to the reliability of its elements, nodes and arcs, will be surveyed. Applications to communication networks will be used as illustrations of the theory.

GENERALIZED FINITE-GEOMETRY CODES

CARLOS R. P. HARTMANN, LUTHER D. RUDOLPH
Systems and Information Science, Syracuse University, Syracuse, N.Y.

ABSTRACT

A technique is presented for constructing cyclic codes that retain many of the combinatorial properties of finite-geometry codes, but are often superior to geometry codes. It is shown that L-step orthogonalization is applicable to certain subclasses of these codes.

1. INTRODUCTION

The use of finite geometries in the construction of cyclic error-correcting codes first appeared in the unpublished work of Prange [1,2], who used the projective planes of orders 4 and 8 to construct and analyze the (21,11) and (73,45) codes respectively. The general classes of projective-geometry and Euclidean-geometry codes were introduced by Rudolph [3,4]. Independently, Weldon [5] introduced difference-set codes, a subclass of the projective-geometry codes. The theory of finite-geometry codes and some generalizations of finite-geometry codes have been further developed by a number of researchers [6-22]. Our purpose in this paper is to present a further generalization of finite Euclidean-geometry and projective-geometry codes.

2. GENERALIZED EUCLIDEAN-GEOMETRY CODES

2.1 Generalized Euclidean geometries

In order to construct a generalized Euclidean geometry, it is first necessary to generalize the concept of "flat". The points of the generalized Euclidean geometry GEG(m,p) over GF(p) will be taken to be the elements of GF(p^m). Thus the points of GEG(m,p) coincide with the points of EG(m,p). The generalized flats of GEG(m,p), which we call "plates", do not in general coincide with the flats of EG(m,p), however. In order to define a plate, it is first necessary to introduce a generalized definition of linear independence.

Let $S_1,\dots,S_k$ be sets of elements from GF(p^m) and let α be a primitive element of GF(p^m). We will say that the points $\alpha^{e_1},\dots,\alpha^{e_k}$ of GEG(m,p) are linearly independent over the sets $S_1,\dots,S_k$ if and only if there is no set of k elements $a_1,\dots,a_k$, not all zero with $a_i \in S_i$ for i = 1,...,k, such that

$$a_1\alpha^{e_1} + \dots + a_k\alpha^{e_k} = 0.$$

Let the positive integer n_j be a proper divisor of $p^m - 1$. Corresponding to each n_j is a proper multiplicative subgroup of the multiplicative group of $GF(p^m)$. Denote by S_j the set of elements of $GF(p^m)$

$$S_j = \{0, 1, \alpha^{\frac{p^m-1}{n_j}}, \alpha^{2\frac{p^m-1}{n_j}}, \ldots, \alpha^{(n_j-1)\frac{p^m-1}{n_j}}\}.$$

Define N_k to be the k-tuple $N_k = (n_1,\ldots,n_k)$, where the positive integers n_j are a set of k proper divisors of $p^m - 1$ with $n_i \leq n_j$ for $i > j$ and $n_j \equiv -1 \pmod p$ for $j = 1,\ldots,k$. We now define a (k,N_k)-plate in $GEG(m,p)$ to be the set of points

$$\alpha^j = \alpha^{e_0} + \beta_1\alpha^{e_1} + \ldots + \beta_k\alpha^{e_k}, \quad \beta_j \in S_j,$$

where $\alpha^{e_1},\ldots,\alpha^{e_k}$ are fixed points in $GEG(m,p)$ that are linearly independent over $S_1,\ldots,S_k$ and β_j ranges over all possible values in S_j, $1 \leq j \leq k$. As in the case of flats of ordinary Euclidean geometries [23], we may represent a plate by a polynomial over $GF(p)$. The term "plate" will be used to denote both the point set and the associated polynomial.

In the special case when $n_j = p^s - 1$ for $j = 1,\ldots,k$, where s is a divisor of m, a (k,N_k)-plate is a k-flat in $EG(\frac{m}{s}, p^s)$. We remark here that if $n_j = p^{s_j} - 1$, where s_j is a divisor of m for $j = 1,\ldots,k$, the (k,N_k)-plate is what Lin and Weldon [18] have called a "frame" in $EG(m,p)$, and if $n_j = p^{s_j} - 1$ for $j = 1,2,\ldots,k$, where n_{j+1} divides n_j for $j = 1,2,\ldots,k-1$, the (k,N_k)-plate is the "generalized flat" introduced by Delsarte [7].

2.2 GEG codes

The $(r,N_{r+1})^{th}$-order generalized Euclidean-geometry (GEG) code of length $n = p^m - 1$ with symbols from $GF(p)$ is defined to be the largest cyclic code whose dual code contains all the $(r+1,N_{r+1})$-plates in $GEG(m,p)$ that do not pass through the origin.

In order to determine the dimension of a GEG code, it is necessary to specify the roots of its parity check polynomial $h(x)$. In order to do this, we require a generalization of the concept of s-weight [23] which we call a "p-cover".

A nonnegative integer t is said to be a p-cover of $N_k = (n_1,\ldots,n_k)$ if and only if there exists a set of integers $b_0, b_1,\ldots,b_k$ satisfying the following conditions:

(i) $t = b_0 + b_1n_1 + \ldots + b_kn_k$ where $b_0 \geq 0$ and $b_i > 0$ for $i = 1,\ldots,k$.

(ii) $t_i \geq \sum_{j=1}^{k} k_{ij}$ for $i = 0,1,\ldots,I$, where t_i and k_{ij} are the i^{th} p-ary digits in the radix -p expansions of t and $b_j n_j$ respectively, i.e.

$$t = t_0 + t_1 p + \ldots + t_I p^I,\ 0 \leq t_i < p$$

$$b_j n_j = k_{0j} + k_{1j} p + \ldots + k_{Ij} p^I,\ 0 \leq k_{ij} < p.$$

For example, let $p = 2$, $m = 6$ and $N_2 = (7,3)$. Then $t = 31 = 011111 = 011000 + 000111$ is a 2-cover of N_2 but $t = 27 = 011011$ is not. In the special case when $n_i = p^s - 1$ for $i = 1,\ldots,k$, t is a p-cover of N_k if and only if $W_s(t) \geq k$, where $W_s(t)$ denotes the s-weight of t.

Theorem 1: Let α be a primitive element of $GF(p^m)$. Then α^t, $0 \leq t < p^m-1$ is a root of $h(x)$, the parity check polynomial of the $(r,N_{r+1})^{th}$-order GEG code, provided that t is not a p-cover of $N_{r+1} = (n_1,\ldots,n_{r+1})$.

The proof of Theorem 1 can be found in [24].

In the general case it is not necessarily true that the intersection of two plates is a plate. This means that in general we cannot determine d_{ML}, the minimum distance guaranteed by L-step orthogonalization. However, in some special cases d_{ML} can be determined, as will be seen in the next section.

2.3 Regular GEG codes

In the special case when n_{j+1} divides n_j for $j = 1,2,\ldots,r$, an $(r+1, N_{r+1})$-plate in GEG(m,p) is called a regular plate. We define a regular $(r,N_{r+1})^{th}$-order GEG code of length $n = p^m - 1$ to be the largest cyclic code whose dual contains all the regular $(r+1,N_{r+1})$-plates in GEG(m,p) that do not pass through the origin where there is at least one regular $(r+1,N_{r+1})$-plate such that $\alpha^{e_1}, \alpha^{e_2},\ldots,\alpha^{e_r}$ are linearly independent over the sets $R_1, R_2,\ldots, R_r$, $R_j = \{\beta_j - \bar{\beta}_j \mid \beta_j, \bar{\beta}_j \in S_j\}$ for $j = 1,2,\ldots,r$.

Theorem 2: The regular $(r,N_{r+1})^{th}$-order GEG code of length $n = p^m - 1$ can be (r+1)-step majority decoded provided $t_{ML} = [(d_{ML} - 1)/2]$ or fewer errors occurred, where

$$d_{ML} = \left\lceil \frac{p^m - |R_1|\ |R_2| \ldots\ |R_r|}{|R_1|\ |R_2| \ldots |R_r|\ n_{r+1}} \right\rceil$$

where [y] denotes the integer part of y, $\lceil x \rceil$ denotes the least integer greater than or equal to x and $|R_j|$ the cardinality of R_j.

We note that $\dfrac{p^m - |R_1|\ |R_2|\ldots|R_r|}{|R_1|\ |R_2|\ldots|R_r|\ n_{r+1}}$ is not necessarily an integer, but for the case $n_j = p^{s_j} - 1$ for $j = 1,2,\ldots,r$,

$$d_{ML} = \frac{p^{m-(s_1+s_2+\ldots+s_r)} - 1}{n_{r+1}} .$$

this result was also obtained by Delsarte [7].

The regular $(r,N_{r+1})^{th}$-order GEG codes for which $n_j = 2^s - 1$ for $j = 1,\ldots,r+1$ are the classical $(r,s)^{th}$-order EG codes. When $n_{r+1} = 1$ $n_j = p^s - 1$ for $j = 1,\ldots,r$, the regular $(r,N_{r+1})^{th}$-order GEG code is a two-fold EG code [18]. We now give some examples of regular GEG codes.

Example 1: The regular $(1,N_2)^{th}$-order GEG code of length $n = 2^6 - 1$ with $N_2 = (3,1)$ is a binary (63,24) code with $t_{ML} = 7$. This code, which was also found by Delsarte [7] and by Lin and Weldon [18], is a BCH code and is orthogonalizable in two steps.

Example 2: The regular $(2,N_3)^{th}$-order GEG code of length $n = 2^{12} - 1$ with $N_3 = (15,1,1)$ is a binary (4095,2000) code with $t_{ML} = 63$. The corresponding $(5,1)^{th}$-order EG code (5^{th} order RM code) with $t_{ML} = 63$ is a binary (4095,1586) code. The GEG code is orthogonalizable in three steps, the EG code in two [25].

Example 3: The regular $(1,N_2)^{th}$-order GEG code of length $n = 2^{16} - 1$ with $N_2 = (15,3)$ is a binary (65535,15715) code with $t_{ML} = 682$. The corresponding $(2,2)^{th}$-order EG code with $t_{ML} = 682$ is a binary (65535, 12273) code. Both codes can be orthogonalized in two steps.

Example 4: The $(0,N_1)^{th}$-order GEG code of length $n = 2^{11} - 1$ with $N_1 = (23)$ is a binary (2047,573) code with $t_{ML} = 44$. This code is 1-step-orthogonalizable.

3. GENERALIZED PROJECTIVE-GEOMETRY CODES

3.1 Generalized projective geometries

Let α be a primitive element of $GF(p^m)$ and n_0 a proper divisor of $p^m - 1$ such that $n_0 \equiv -1 (\mathrm{mod}\ p)$. The sets of n_0^{th} roots of unity form a proper subgroup, G, of the multiplicative group of $GF(p^m)$. The points of the generalized projective geometry $GPG(m,n_0,p)$ over GF(p) will be taken to be the cosets with respect to G in the multiplicative group of GF(p^m). The coset $\{\alpha^{j_1},\dots,\alpha^{j_{n_0}}\}$ will be denoted by (α^j) where $j = \min(j_1\dots,j_{n_0})$. Note that under this index convention the cosets are (α^0), (α^1),..., (α^{n-1}), where $n = (p^m - 1)/n_0$.

Let $N_k = (n_1,\dots,n_k)$ where the positive integers n_j are a set of k proper divisors of $p^m - 1$, with $n_i \le n_j$ for $i > j$, $n_j = \theta_j n_0$, and $n_j \equiv -1 (\mathrm{mod}\ p)$ for $j = 1,\dots,k$. Denote by S_j the set of elements

$$S_j = \{0,1,\alpha^{\frac{p^m-1}{n_j}}, \alpha^{2\frac{p^m-1}{n_j}},\dots,\alpha^{(n_j-1)\frac{p^m-1}{n_j}}\}$$

for j = 0,.,...,k. We define a (k,N_k)-plate in GPG(m,n_0,p) to be the set of points

$$(\alpha^j) = (\beta_0\alpha^{e_0} + \dots + \beta_k\alpha^{e_k}),\ \beta_i \in S_i \text{ and } \beta_i \text{ not all } 0,$$

where $\alpha^{e_0},\dots,\alpha^{e_k}$ are a fixed set of k+1 points of GEG(m,p) that are linearly independent over the sets $S_0,\dots,S_k$, and β_j ranges over all possible values in S_j except that not all β_i are simultaneously zero. As in the case of flats in an ordinary finite projective geometry [23], we may represent a plate in GPG(m,n_0,p) by a polynomial of degree less than n.

In the special case where $n_j = p^s - 1$ for $j = 0,1,\dots,k$, a (k,N_k)-plate in GPG(m,n_0,p) is a k-flat in PG$((m-s)/s,p^s)$. We remark here that if $n_j = p^{s_j} - 1$ for $j = 0,1,\dots k$, where n_{j+1} divides n_j for $j = 0,1,\dots,k-1$, the (k,N_k)-plate is the "generalized flat" introduced by Delsarte [7].

3.2 GPG codes

The $(r,N_r)^{th}$-order generalized projective-geometry (GPG) code of length $n = (p^m - 1)/n_0$ with symbols from GF(p) is defined to be the largest cyclic code whose dual code contains all the (r,N_r)-plates in GPG(m,n_0,p).

The roots of the parity check polynomial h(x) of a GPG code are specified by the following

Theorem 4: Let α be a primitive element of GF(p^m). Then α^{tn_0}, $1 \le t < n$, is a root of h(x), the parity check polynomial of the $(r,N_r)^{th}$-order GPG code, provided that tn_0 is not a p-cover of $N_{r+1} = (n_1,\dots,n_r, n_{r+1})$, where $N_r = (n_1,n_2,\dots,n_r)$ and $n_{r+1} = n_0$.

The proof of Theorem 4 can be found in [24].

As was the case for generalized Euclidean geometries, the intersection of two plates in a generalized projective geometry is not necessarily a plate, so that we cannot in general calculate d_{ML} for GPG codes. There is a special case, however, for which d_{ML} can be determined.

3.3 Regular GPG codes

In the special case where n_{j+1} divides n_j for $j = 1,\ldots,r-1$, an (r,N_r)-plate in $GPG(m,n_0,p)$ is called a regular plate. We define a regular $(r,N_r)^{th}$-order GPG code of length $n = (p^m - 1)/n_0$ to be the largest cyclic code whose dual code contains all the regular (r,N_r)-plates in $GPG(m,n_0,p)$ where there is at least one regular (r,N_r)-plate such that $\alpha^{e_0}, \alpha^{e_1},\ldots, \alpha^{e_{r-1}}$ are linearly independent over the sets $R_0,R_1,\ldots,R_{r-1}$, $R_j = \{\beta_j-\bar\beta_j \mid \beta_j,\bar\beta_j \in S_j\}$ for $j = 0,1,\ldots,r-1$.

Theorem 5: The regular $(r,N_r)^{th}$-order GPG code of length $n = (p^m - 1)/n_0$ can be r-step majority decoded provided that $t_{ML} = [(d_{ML} - 1)/2]$ or fewer errors occurred, where $d_{ML} = \min_{0\le j<r} \{\bar J_{r-j+1} + 1\}$ and

$$\bar J_{r-j+1} = \left\lceil \frac{p^m - |R_1||R_2|\ldots|R_{r-j}|}{|R_1||R_2|\ldots|R_{r-j}|\; n_{r-j}} \right\rceil$$

We note that $\dfrac{p^m - |R_1||R_2| \ldots |R_{r-j}|}{|R_1||R_2| \ldots |R_{r-j}|\; n_{r-j}}$ is not necessarily an integer, but for the case where $n_j = p^{s_j} - 1$ for $j = 1,2,\ldots,r$,

$$\bar J_{r-j+1} = J_{r-j+1} + 1 = \frac{p^{m-(s_1+\ldots+s_{r-j})} - 1}{p^{s_{r-j}} - 1} .$$

This result was also obtained by Delsarte [7].

The regular $(r,N_r)^{th}$-order codes for which $n_j = 2^s - 1$ for $j = 0,1,\ldots r$ are the classical $(r,s)^{th}$-order PG codes. In this case $d_{ML} = \bar J_{r+1} + 1$. We now give two examples of regular GPG codes.

Example 5: The regular $(2,N_2)^{th}$-order GPG code of length $n = (2^{16} - 1)/3$ with $n_0 = 3$ and $N_2 = (15,3)$ is a binary (21845,8908) code with $t_{ML} = 136$. The corresponding PG code is the $(3,2)^{th}$-order

(21845,8536) code with $t_{ML} = 170$. Both codes can be majority decoded in two steps [25].

Example 6: The regular $(3,N_3)^{th}$-order GPG code of length $n = (2^{20} - 1)/3$ with $n_0 = 3$ and $N_3 = (15,3,3)$ is a binary (349525,145859) code with $t_{ML} = 682$. The corresponding PG code with $t_{ML} = 682$ is the $(4,2)^{th}$-order (349525,145055) code. The GPG code can be majority decoded in three steps, the PG code in two.

We remark that L-step orthogonalization is not the best decoding algorithm for all regular generalized finite geometry codes as can be seen in the next example.

Example 7: The uniform $(1,N_1)^{th}$-order GPG code of length $n = (2^8 - 1)/5$ with $n_0 = n_1 = 5$ is a binary (51,16) code with minimum distance $d = 16$ [26] and BCH bound $d_{BCH} = 12$. By Theorem 5 $d_{ML} = 4$, but using 49 nonorthogonal $(1,N_1)$-plates in GPG(8,5,2) it is possible to correct up to six errors [27] by one step weighted-majority decoding [28].

4. DISCUSSION

We have presented a technique for constructing cyclic codes that retain many of the combinatorial properties of finite-geometry codes, but which are in many cases superior to these codes. We have been able to show that L-step orthogonalization is applicable to some of these new codes. For others, weighted-majority decoding using nonorthogonal parity checks is more appropriate. Because of their rich subcode structure, generalized finite-geometry codes are particularly well suited for decoding by sequential code reduction. This makes generalized finite-geometry codes attractive for use in practical error-control systems where very long codes are required.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contributions to this work by graduate students Ralph Longobardi and James Ducey of Systems and Information Science, Syracuse University.

REFERENCES

1. Prange, E., "Some Cyclic Error Correcting Codes with Simple Decoding Algorithms," AFCRC-TN-58-156, Air Force Cambridge Research Center, Cambridge, Mass. (1958).

2. Prange, E., "The Use of Coset Equivalence in the Analysis and Design of Group Codes," AFCRC-TR-59-164, Air Force Cambridge Research Center, Cambridge, Mass. (1959).

3. Rudolph, L. D., "Geometric Configurations and Majority Logic Decodable Codes," MEE Thesis, University of Oklahoma, Norman, Oklahoma (1964).

4. Rudolph, L. D., "A Class of Majority Logic Decodable Codes," IEEE Trans. on Info. Theory, IT-13, pp. 305-307 (1967).

5. Weldon, E. J., Jr., "Difference-Set Cyclic Codes," Bell System Tech. J., 45, pp. 1045-1055 (1966).

6. Chow, D. K., "A Geometric Approach to Coding Theory with Application to Information Retrieval," Coordinated Sci. Lab. Rept. R-368, University of Illinois, Urbana (1967).

7. Delsarte, P., "A Geometric Approach to a Class of Cyclic Codes," Journal of Combinatorial Theory, 6, pp. 340-358 (1969).

8. Delsarte, P., "Majority Logic Decodable Codes Derived from Finite Inversive Planes," Information and Control, 16, pp. 403-442 (1970).

9. Delsarte, P., J. M. Goethals, and F. J. MacWilliams, "On GRM and Related Codes," Information and Control, 16, pp. 403-442 (1970).

10. Goethals, J. M. and P. Delsarte, "On a Class of Majority-Logic Decodable Codes," IEEE Trans. on Info. Theory, IT-14, pp. 182-188 (1968).

11. Graham, F. L. and F. J. MacWilliams, "On the Number of Parity Checks in Difference Set Cyclic Codes," Bell System Tech. J., 45, pp. 1057-1070 (1966).

12. Hamada, N., "The Rank of the Incidence Matrix of Points of d-flats in Finite Geometries," J. Sci. Hiroshima University, 32, pp. 381-396 (1968).

13. Kasami, T., S. Lin and W. W. Peterson, "New Generalizations of the Reed-Muller Codes - Part I: Primitive Codes," IEEE Trans. on Info. Theory, IT-14, pp. 189-198 (1968).

14. Kasami, T., S. Lin and W. W. Peterson, "Polynomial Codes," IEEE Trans. on Info. Theory, IT-14, pp. 807-814 (1968).

15. Lin, S., "On a Class of Cyclic Codes," Chapter 7, Error-Correcting Codes, H. Mann, Ed., Wiley, New York (1968).

16. Lin, S., "On the Number of Information Symbols of Polynomial Codes," IEEE Trans. on Info. Theory, to appear.

17. Lin, S., "Multifold Euclidean Geometry Codes," to appear.

18. Lin, S. and E. J. Weldon, Jr., "New Efficient Majority-Logic-Decodable Cyclic Codes," presented at the IEEE International Symposium on Info. Theory, Asilomar, California (1972).

19. MacWilliams, F. J. and H. B. Mann, "On the p-rank of the Design Matrix of a Difference Set," Information and Control, 12, pp. 474-488 (1968).

20. Smith, K. J. C., "On the p-rank of the Incidence Matrix of Points and Hyperplanes in a Finite Projective Geometry," J. Combinatorial Theory, 7, pp. 122-129 (1969).

21. Weldon, E. J., Jr., "Euclidean Geometry Cyclic Codes," Proceedings of the Symposium of Combinatorial Mathematics at the University of North Carolina, Chapel Hill, N.C. (1967).

22. Weldon, E. J., Jr., "New Generalizations of the Reed-Muller Codes - Part II: Non-primitive Codes," IEEE Trans. on Info. Theory, IT-14, pp. 199-205 (1968).

23. Peterson, W. W. and E. J. Weldon, Jr., Error-Correcting Codes, 2nd Edition, M.I.T. Press, Cambridge, Mass. (1972).

24. Hartmann, C.R.P. and L. D. Rudolph, "Generalized Finite-Geometry Codes," Systems and Information Science Technical Report, Syracuse University, Syracuse, New York (1972).

25. Chen, C. L., "Note on Majority-Logic Decoding of Finite Geometry Codes," IEEE Trans. on Info. Theory, to appear.

26. Berlekamp, E. R., Algebraic Coding Theory, McGraw-Hill, New York (1968).

27. Ducey, J., private communication (April, 1972).

28. Rudolph, L. D. and W. E. Robbins, "One-Step Weighted-Majority Decoding", IEEE Trans. on Info. Theory, IT-18, pp. 446-448 (1972).

CODING THEORY AND THE COMPUTER

Robert T. Chien
Coordinated Science Laboratory
University of Illinois at
Urbana-Champaign

ABSTRACT

The application of coding to computer systems is examined in some detail with the help of several illustrative examples of actual applications in systems. It becomes evident from these examples that many useful systems with coding has been developed and the success of these systems is a result of taking intȯ account of not only the details of coding and decoding but also the special system evironment in which they operate.

Coding theory has been applied to enhance the reliability of computer systems in many different ways. Depending on the specific subsystem which employs coding techniques, the nature of encoding and decoding varies greatly. In general, the CPU due to its relatively high reliability in computers employs only error detection in the form of a residual check. The most common form used are residues modulo 3, 7, and 15. The modulo 3 check will detect any single error in computer arithmetic; while the other two checks will detect combinations of multiple faults. Implementation for error detection can be simply accomplished without loss of speed or capacity. In the meantime, many advanced residue codes have been proposed for error correcting. This includes single correcting codes by Brown [1], large distance codes by Barrows [2] and Mandelbaum [3] and other classes of codes which can be decoded with majority decoding [4]. However, the codes proposed require either a significant reduction where comparison or a significant amount of hardware implementation. Further proposals may be necessary before wide application is anticipated.

The high speed core memory provides an approach for the application of coding. The problem here is relatively simple as errors occur independently of each other. As a result random error correcting codes can be applied in a straightforward manner. The major concern here is one of cost as well as decoding time. Any decoding that significantly increases the cycle time of the core memory will not be looked upon favorably by practicing engineers.

By and large, the greatest gain in reliability and capacity with the use of coding lies in the area of mass memory. This includes magnetic tape, disc files and other large scale mass memory such as high density optical storage systems.

In the disc file the errors are bursty and due primarily to impurities on the disc. This is an ideal situation for the application of burst correcting codes which are highly efficient and easy to implement. The only drawback with existing burst correcting codes lies in the need of decoding time which is of a comparable magnitude of the time to read data. For any application where the records on the disc file are very long, namely on the order of 50,000 characters, the decoding would become unbearable. An important version of the Fire code with multiple factor and high speed decoding algorithm seem to have resulted in a practical system for the disc file [5].

An interesting application of coding was developed for the magnetic tape system which makes excellent use of the existing redundancy in the form of a parity check. In the tape series used in System/360, a simple coding scheme was developed to correct unlimited types of errors as long as they are confined to a single track. This scheme by Brown and Sellers [6] utilizes a check character called a Cyclic Redundancy Check (CRC). The old Longitudinal Redundancy Check (LRC) is not used in the system. The CRC is a polynomial code which employs a code which is reciprocal, therefore, allowing error detection in both directions. In encoding each character on the tape is shifted back through a feedback shift register whose leads are determined by the polynomial. A shift to the right is made every time a new character is entered. A zero character is inserted as the last data block which generates the final register content as the CRC of the record. The CRC is then recorded on the tape as the last character of the record. During the decoding the same process is followed. Each character is shifted through a CRC register including the necessary character which is a CRC. If no errors have occurred the register count should be zero. If the register count at the end is not zero an error has been detected. Error correction is achieved with the help of another register, the Error Pattern Register (EPR). The connections of the EPR is the same as the CRC register. The only input to the EPR is through the leading stage. A "one" is read into EPR if and only if the Vertical Redundancy Check (VRC) fails. The content of the final CRC register is shifted to match the final character of the EPR. The number of shifts necessary to achieve a match indicates the track which contains the error.

This application indicates a judicial combination of coding theory and the use of existing redundancy in the system. The system is actually quite powerful because it is able to correct new error patterns within a single track which is not divisible by the CRC polynomial. It also is able to detect multiple track errors.

Another example of a major application of coding to mass memory is a large scale optical memory which has a capacity of 10^{12} bits. In this system bits are recorded on an optical chip which contains large numbers of track of data. The noise environment is such that short multiple bursts dominate. A large fraction of the errors are indeed single errors. The initial error rates are of the order of 10^{-3}. To achieve a system reliability of 10^{-8} it was necessary to use a Reed-Soloman code which is based on a 6 bit character and a bit length 6 X 63 = 378 bits. Using 11 characters of redundancy which amounts to 17%, it is possible to correct five character errors and detect 6. The computer system environment dictates the use of very fast decoding algorithms. To achieve this, encoding and parity check are done by hardware. A fast decoding algorithm which corrects one character of error is implemented as a fast subroutine resident in core. The decoding then is approximately 0.3 millisecond which is within the time of one scan of a single track. After the attempted correction of a single error character the parity is applied once more for verification. The single error correcting algorithm can be applied as many as ten times with rescan of the same data. Multiple error correcting algorithms are programmed and they are used only after the single error correcting algorithm has failed at least ten times. These programs are stored on the disc file and therefore require more time for execution. The system works very well with the hybrid combination for fast parity checking and single error correcting and back-up software oriented toward multiple error correction. The reliability improvement was about five orders of magnitude with a throughput degradation of only 0.7%. This system is another example of the application of coding which is economical and has merit from the point of view of cost effectiveness.

REFERENCES

[1] D. T. Brown, "Error Detecting and Correcting Binary Codes for Arithmetic Operations," IRE Trans., EC-9, pp. 333-337, 1960.

[2] J. T. Barrows, Jr., A New Method for Constructing Multiple Error Correcting Linear Residue Codes, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, Report R-277.

[3] D. Mandelbaum, "Arithmetic Codes with Large Distance," IEEE Trans., IT-13, No. 2, pp. 237-242, 1967.

[4] C. L. Chen, R. T. Chien, and C. K. Liu, "On a Class of Majority-Logic Decodable Arithmetic Codes," 1972 Int. Symp. on Information Theory, Asilomar, California, January 1972.

[5] R. T. Chien, "Burst-Correcting Codes with High-Speed Decoding," IEEE Transactions on Information Theory, Vol. IT-15, Number 1, pp. 109-113, January 1969.

[6] D. T. Brown and F. F. Sellers, Jr., "Error Correction for IBM 800-bit-per-inch Megnetic Tape," IBM J. Res. Develop., Vol. 14, No. 4, pp. 341-474, July 1970.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant GK-24879 and in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy and U. S. Air Force) under Contract DAAB-07-72-C-0259.

COMPUTATION BOUNDS ON 3-GROUP BOOTSTRAP DECODING

DANIEL J. COSTELLO, JR.
Illinois Institute of Technology
Chicago, Illinois

FREDRICK JELINEK
Cornell University
Ithaca, New York

ABSTRACT

A description is given for an extension of the bootstrap hybrid decoding algorithm from a single parity check to two parity checks. The resultant code is called a 3-group code because the digits can be divided into 3 groups with similar properties. An upper bound on the moments of the decoding effort for this scheme is then derived and the R_{comp} of the 3-group algorithm is compared to the R_{comp} of the single parity check algorithm.

INTRODUCTION

A thorough familiarity with the original paper on bootstrap hybrid decoding by Jelinek and Cocke [1] will be assumed. Due to space limitations, most of the arguments will just be highlighted. Those interested in more detail are referred to [2]. The binary symmetric channel (BSC) will be assumed throughout.

THREE GROUP BOOTSTRAP DECODING

Description of Code and Its Use in Bootstrapping

It is desirable to generalize bootstrap decoding to encode transmitted streams by use of an algebraic code that has more than one parity check. The three-group code has two parity check digits v_1, v_2 and k information digits $m_1,\dots,m_k$. Every information digit is checked by at least one parity check digit. Without loss of generality let

$$v_1 = \sum_{i=1}^{\ell-1} m_i \qquad 1 \le \ell - 1 \le k$$
$$v_2 = \sum_{i=h}^{k} m_i \qquad 1 \le h \le k \tag{1}$$

For the code to be non-trivial, $1 \le h \le \ell - 1 \le k$ and at least one of the outside inequalities is strict. It is convenient to define the codeword digits, $x_1,\dots,x_{k+2}$ as follows:

$$x_1 = v_1$$
$$x_i = m_{i-1} \qquad i = 2,\dots,k+1 \tag{2}$$
$$x_{k+2} = v_2$$

The codeword digits may then be divided into three groups

$$S_1 = \{x_1,\ldots,x_h\}\ ,\quad S_2 = \{x_{h+1},\ldots,x_\ell\},\quad S_3 = \{x_{\ell+1},\ldots,x_n\} \tag{3}$$

where n=k+2. Let $\underline{y} = y_1,\ldots,y_n$ be the received digits, and define

$$\begin{aligned} t_1 &\triangleq \sum_{i=1}^{h} y_i & u_1 &\triangleq \sum_{i=1}^{h} x_i \\ t_2 &\triangleq \sum_{i=h+1}^{\ell} y_i & u_2 &\triangleq \sum_{i=h+1}^{\ell} x_i \\ t_3 &\triangleq \sum_{i=\ell+1}^{n} y_i & u_3 &\triangleq \sum_{i=\ell+1}^{n} x_i \end{aligned} \tag{4}$$

Then the syndrome digits of $\underline{y}$ are

$$s_1 = t_1 \oplus t_2 \quad \text{and} \quad s_2 = t_2 \oplus t_3 \tag{5}$$

Let $\underline{u} = (u_1,u_2,u_3)$, $\underline{t} = (t_1,t_2,t_3)$, $n_1 = h$, $n_2 = \ell-h$, $n_3 = n-\ell$. Assuming that the information digits $m_1,\ldots,m_k$ are independent and identically distributed with $P(m_i=0) = P\{m_i=1\} = 1/2$, then

$$P\{\underline{u} = 0,0,0\} = P\{\underline{u} = 1,1,1\} = 1/2 \tag{6}$$

Now for $n_i \geq 1$,

$$q_{n_i}(0) \triangleq P\{t_i = u_i \mid u_i\} = P\left\{\begin{array}{l}\text{an even number of } n_i \\ \text{digits were received} \\ \text{incorrectly through} \\ \text{the channel}\end{array}\right\} = \frac{1+(1-2p)^{n_i}}{2} \tag{7a}$$

where p is the channel crossover probability. As a consequence,

$$q_{n_i}(1) \triangleq P\{t_1 \neq u_i \mid u_i\} = \frac{1-(1-2p)^{n_i}}{2}\ . \tag{7b}$$

It will prove convenient to also define

$$q_o(0) \triangleq 1 \quad , \quad q_o(1) \triangleq 0\ . \tag{8}$$

From the above, we get the relation

$$\begin{aligned} P\{t_1,t_2,t_3\} = \tfrac{1}{2}\,[\,& q_{n_1}(t_1)\ q_{n_2}(t_2)\ q_{n_3}(t_3)\ + \\ & q_{n_1}(t_1 \oplus 1)\ q_{n_2}(t_2 \oplus 1)\ q_{n_3}(t_3 \oplus 1)] \end{aligned} \tag{9}$$

where $\oplus$ denotes mod 2 summation, and $n_j \geq 1$.

It can be shown that for the three group code,

$$\frac{P\{\underline{y}/x_i\}}{P\{\underline{y}\}} = \frac{P\{y_i,t_1,t_2,t_3/x_i\}}{P\{y_i,t_1,t_2,t_3\}} \quad \text{if } n_j \geq 1\ . \tag{10}$$

Since the left-hand ratio is the one that enters into the likelihood calculation for bootstrap decoding, the receiver will be interested in probabilities $P\{y_i, t_1, t_2, t_3/x_i\}$.

After some manipulation it can be shown that as long as $n_1 \geq 1$, $n_2 \geq 1$, $n_3 \geq 1$, then

$$P\{y_i, \underline{t}/x_i\} = \frac{1}{2} w(y_i/x_i) \left[q_{n_1-1}(t_1 \oplus y_i \oplus x_i)\, q_{n_2}(t_2)\, q_{n_3}(t_3) + q_{n_1-1}(t_1 \oplus y_i \oplus x_i \oplus 1)\, q_{n_2}(t_2 \oplus 1)\, q_{n_3}(t_3 \oplus 1) \right], \tag{11}$$

where $w(y_i/x_i)$ is the probability of receiving y_i given that x_i was transmitted. As a consequence,

$$P\{y_i, \underline{t}\} = \frac{1}{4} \left[q_{n_1}(t_1)\, q_{n_2}(t_2)\, q_{n_3}(t_3) + q_{n_1}(t_1 \oplus 1)\, q_{n_2}(t_2 \oplus 1)\, q_{n_3}(t_3 \oplus 1) \right], \tag{12}$$

so that for $n_1 \geq 1$, $n_2 \geq 1$, $n_3 \geq 1$,

$$\frac{P\{y_i, \underline{t}/x_i\}}{P\{y_i, \underline{t}\}} = 2w(y_i/x_i) \frac{q_{n_1-1}(t_1 \oplus y_i \oplus x_i)\, q_{n_2}(t_2)\, q_{n_3}(t_3) + q_{n_1-1}(t_1 \oplus y_i \oplus x_i \oplus 1)\, q_{n_2}(t_2 \oplus 1)\, q_{n_3}(t_3 \oplus 1)}{q_{n_1}(t_1) q_{n_2}(t_2) q_{n_3}(t_3) + q_{n_1}(t_1 \oplus 1) q_{n_2}(t_2 \oplus 1) q_{n_3}(t_3 \oplus 1)} . \tag{13}$$

Relation (13) was obtained under the assumption that $i \varepsilon [1, \ldots, n_1]$. If $n_1 + 1 \leq i \leq n_1 + n_2$, we need only interchange n_1 and t_1 with n_2 and t_2 in (13). The interchange of n_1 and t_1 with n_3 and t_3 preserves the validity of (13) for $n_1 + n_2 + 1 \leq i \leq n$.

It follows from (9), (10), and (13) that if $n_i \geq 1$, the likelihood used in bootstrap decoding with a three group algebraic code is a function of y_i, x_i, and the state variables $(t_1, t_2, t_3, n_1, n_2, n_3)$. We will see that these variables will also be sufficient if all the digits of one or two of the three groups have been decoded. The needed adjustment of the state variable values as the decoding proceeds is as follows:

At the beginning, when no digit in a column has yet been decoded, (13) is used directly. Suppose, without loss of generality (w.l.o.g.), that y_1 is decoded as x_1. Then a new $t_1' = t_1 \oplus x_1 \oplus y_1$ is obtained and used in (13) with n_1 replaced by $n_1 - 1$. This process continues until all digits of some set S_i have been decoded. W.l.o.g. assume that such a set is S_2, that the new t-values are t_1, t_2, t_3 and that n_1 and n_3 digits remain undecoded in S_1 and S_3. Assuming that no error was committed, $t_2 = u_2$, and when decoding y_i for $1 \leq i \leq n_1$ the value of t_3 becomes irrelevant and only those of t_1 and t_2 count. When $n_1 = 1$, the remaining y_i can be decoded algebraically from the relation

$$x_i = t_1 \oplus t_2 \oplus y_i = t_2 \, . \tag{14}$$

We can show that the ratio (13) remains valid even if one of the groups is completely decoded. In fact, if $n_1 = 1$ and $n_2 = 0$, (13) becomes (note that $t_1 = y_i$ if $n_1 = 1$)

$$\frac{P\{y_i, \underline{t}/x_i\}}{P\{y_i, \underline{t}\}} = 2w(y_i/x_i) \; \frac{q_0(t_1 \oplus t_2 \oplus y_i \oplus x_i)}{q_1(t_1 \oplus t_2)} = \begin{cases} 2 & \text{if } x_i = t_2 \\ 0 & \text{if } x_i \neq t_2 \end{cases} \qquad (15)$$

so that straight-forward sequential decoding using the likelihood function based on (13) will force the decoder to select the path on which (14) is satisfied at each depth.

As seen from above, the value of t_3 is irrelevant once all digits of S_2 were decoded and those of S_1 are being decoded. Of course when the latter task is complete, decoding of S_3 starts that will depend on t_3 and t_2 (note that since $u_1 \oplus u_2 = 0$ then $t_1 = t_2$ when S_1 and S_2 have been decoded) in the same way that the just described decoding was dependent on t_1 and t_2.

AN UPPER BOUND ON THE MOMENTS OF THE DECODING EFFORT

First of all, assume there are n_i streams left to be decoded in group i, i = 1,2,3. Then let $N_i(\underline{n})$ be the number of steps necessary to decode any given stream in group i when the step allocation is M = 1 and $\underline{n} = (n_1, n_2, n_3)$. Applying well know results about ordinary sequential decoding, we can conclude that

$$P\left\{N_i(\underline{n}) > \ell\right\} \leqq K(R,\nu)(\Gamma + t)\ell^{-\sigma_i} \qquad (16)$$

where $\Gamma + t$ is the length of the information sequence and $K(R,\nu)$ is a function of rate R and constraint length ν which is finite if ν is finite and σ_i satisfies

$$R < \frac{E_{\underline{n}}(\sigma_i)}{\sigma_i} \qquad \text{for } R \geqq \frac{E_{\underline{n}}(2)}{2}$$

or $\qquad (17)$

$$R < \frac{E_{\underline{n}}(2)}{\sigma_i} \qquad \text{for } R \leqq \frac{E_{\underline{n}}(2)}{2} \quad .$$

In (17), $E_{\underline{n}}(\sigma_3)$ is the concave, positive, increasing function of σ_3 defined as follows:

Let $k = n_1 + n_2 + n_3$ and label the k received digits left to be decoded as $y_1, y_2, \ldots, y_{n_1}, y_{n_1+1}, \ldots, y_{n_1+n_2}, y_{n_1+n_2+1}, \ldots, y_{n_1+n_2+n_3} = y_k$. Define $\underline{y} = (y_1, \ldots, y_k)$ and $\underline{\tilde{y}} = (y_1, \ldots, y_{k-1})$ and assume that the k^{th} stream is in group 3 to be decoded. Then

$$E_{\underline{n}}(\sigma_3) = 1 + \sigma - \log \left[\sum_{\underline{y}} \left(\sum_{x_k=0}^{1} P(\underline{y}|x_k)^{1/1+\sigma} \right)^{1+\sigma} \right] \quad . \qquad (18)$$

From the analysis of the three group code, we obtain the rather long but straight-forward expression

$$
\begin{aligned}
E_{\underline{n}}(\sigma_3) = \sigma_3 - \log \Biggl[& \biggl(\Bigl\{(1-p)\bigl[q_{n_1}(0)q_{n_2}(0)q_{n_3-1}(0)+q_{n_1}(1)q_{n_2}(1)q_{n_3-1}(1)\bigr]\Bigr\}^{1/1+\sigma_3} \\
& + \Bigl\{p\bigl[q_{n_1}(0)q_{n_2}(0)q_{n_3-1}(1)+q_{n_1}(1)q_{n_2}(1)q_{n_3-1}(0)\bigr]\Bigr\}^{1/1+\sigma_3} \biggr)^{1+\sigma_3} \\
& + \biggl(\Bigl\{(1-p)\bigl[q_{n_1}(0)q_{n_2}(0)q_{n_3-1}(1)+q_{n_1}(1)q_{n_2}(1)q_{n_3-1}(0)\bigr]\Bigr\}^{1/1+\sigma_3} \\
& + \Bigl\{p\bigl[q_{n_1}(0)q_{n_2}(0)q_{n_3-1}(0)+q_{n_1}(1)q_{n_2}(1)q_{n_3-1}(1)\bigr]\Bigr\}^{1/1+\sigma_3} \biggr)^{1+\sigma_3} \\
& + \biggl(\Bigl\{(1-p)\bigl[q_{n_1}(1)q_{n_2}(0)q_{n_3-1}(0)+q_{n_1}(0)q_{n_2}(1)q_{n_3-1}(1)\bigr]\Bigr\}^{1/1+\sigma_3} \\
& + \Bigl\{p\bigl[q_{n_1}(1)q_{n_2}(0)q_{n_3-1}(1)+q_{n_1}(0)q_{n_2}(1)q_{n_3-1}(0)\bigr]\Bigr\}^{1/1+\sigma_3} \biggr)^{1+\sigma_3} \\
& + \biggl(\Bigl\{(1-p)\bigl[q_{n_1}(1)q_{n_2}(0)q_{n_3-1}(1)+q_{n_1}(0)q_{n_2}(1)q_{n_3-1}(0)\bigr]\Bigr\}^{1/1+\sigma_3} \\
& + \Bigl\{p\bigl[q_{n_1}(1)\, q_{n_2}(0)q_{n_3-1}(0)+q_{n_1}(0)q_{n_2}(1)q_{n_3-1}(1)\bigr]\Bigr\}^{1/1+\sigma_3} \biggr)^{1+\sigma_3} \Biggr] \qquad (19)
\end{aligned}
$$

where p is the channel crossover probability and

$$
q_{n_i}(0) = \frac{1+(1-2p)^{n_i}}{2} \qquad\qquad q_{n_i}(1) = \frac{1-(1-2p)^{n_i}}{2} \quad . \qquad (20)
$$

Clearly, $E_{\underline{n}}(\sigma_1)$ and $E_{\underline{n}}(\sigma_2)$ are defined in a similar way.

Next let $\sigma_i(\underline{n})$ be the least upper bound on the numbers σ_i satisfying (17), i.e., $\sigma_i(\underline{n})$ is the solution of

$$
\begin{aligned}
R &= \frac{E_{\underline{n}}(\sigma_i(\underline{n}))}{\sigma_i(\underline{n})} && \text{for} \quad \frac{E_{\underline{n}}(2)}{2} \leqq R < C \\
\sigma_i(\underline{n}) &= \frac{E_{\underline{n}}(2)}{R} && \text{for} \quad 0 < R < \frac{E_{\underline{n}}(2)}{2} \quad . \qquad (21)
\end{aligned}
$$

Now choose k(R) to be the largest positive integer such that

$$
k(R)\, \sigma(\infty) < \min_{N} \Bigl\{ \max \bigl[\sigma_1(\underline{n}),\, \sigma_2(\underline{n}),\, \sigma_3(\underline{n})\bigr] \Bigr\} \qquad (22)
$$

where $N = \{\underline{n} = (n_1,n_2,n_3) \mid n_1+n_2+n_3 = k(R)\}$ and $\sigma(\infty)$ is the Pareto exponent which would be obtained with ordinary sequential decoding, i.e., $\sigma(\infty)$ is the solution of

$$
\begin{aligned}
R &= \frac{E_\infty(\sigma(\infty))}{\sigma(\infty)} && \text{for} \quad \frac{E_\infty(2)}{2} \leqq R < C \\
\sigma(\infty) &= \frac{E_\infty(2)}{R} && \text{for} \quad 0 < R < \frac{E_\infty(2)}{2}
\end{aligned} \qquad (23)
$$

where

$$E_\infty(\sigma) = \sigma - \log \left\{ (1-p)^{1/1+\sigma} + p^{1/1+\sigma} \right\}^{1+\sigma} \quad . \tag{24}$$

If there are originally m streams of digits to decode, we wish to modify the three group bootstrap decoding algorithm as follows:

(1) Decode m-k(R) streams by ordinary sequential decoding without the help of the two parity streams and with the step allocation M=1.

(2) Decode the remaining k(R) streams with the help of the parity streams using the three group bootstrap decoding algorithm.

In part (1) of the modified algorithm, the easiest m-k(R) streams are decoded by ordinary sequential decoding. If L^* is the number of steps needed to decode the hardest of the decoded streams, then $P(L^* > \ell)$ is upper bounded by the probability that there is a set of k(R) + 1 streams that need more than ℓ steps each to decode by ordinary sequential decoding. Since the decoding of the first m-k(R) streams is independent, the γth computational moment of the decoding effort in part (1) is bounded if $(k(R)+1)\,\sigma(\infty) > \gamma$.

In part (2) of the modified algorithm, we compute the three Pareto exponents $\sigma_1(\underline{n})$, $\sigma_2(\underline{n})$, and $\sigma_3(\underline{n})$ given that decoding starts in group 1, group 2, or group 3. We then begin decoding in the group with the largest exponent. After decoding each stream, this procedure is repeated, thereby assuring that each successive stream is easier to decode than the previous one. If L(k(R)) is the number of steps needed to decode at least one of the k(R) remaining streams, then $P[L(k(R))] > \ell$ is upper bounded by the probability that there is a set of k(R) streams that need more than ℓ steps each to decode by the three group bootstrap decoding algorithm. Since the decoding of the last k(R) streams is not independent, the γth computational moment of the decoding effort is part (2) is bounded if $\max\left[\sigma_1(\underline{n}),\sigma_2(\underline{n}),\sigma_3(\underline{n})\right] > \gamma$.

In bounding the decoding effort for the complete modified algorithm, we must consider the fact that after the first m-k(R) streams have been decoded any of the situations in the set N may describe the distribution of the remaining k(R) streams. Since in part (1), we decode the m-k(R) easiest streams, we are not free to choose the situation which would give us the best Pareto exponent for part (2). Hence the worst case must be assumed, and the bounding condition in part (2) minimized over all situations in N.

Finally, since the decoding effort must be bounded for both part (1) and part (2), the γth computational moment of the decoding effort is bounded if $\min\left\{(k(R)+1)\sigma(\infty),\ \min\left[\max(\sigma_1(\underline{n}),\ \sigma_2(\underline{n}),\ \sigma_3(\underline{n}))\right]\right\} > \gamma$. We can now summarize as follows:

Theorem: The modified three group bootstrap decoding algorithm leads to a finite γth moment of computation per decoded digit if

$$\min\left\{(k(R)+1)\sigma(\infty),\ \min_{N}\left[\max(\sigma_1(\underline{n}),\ \sigma_2(\underline{n}),\ \sigma_3(\underline{n}))\right]\right\} > \gamma \tag{25}$$

where k(R) is the unique integer satisfying (22), $\sigma(\infty)$ is the unique solution of (23), and $\sigma_i(\underline{n})$ is the unique solution of (21), i = 1, 2, 3.

It is necessary to derive the above bound in terms of a modified decoding algorithm due to the difficulties involved in taking the dependencies of the bootstrap algorithm into account. It should also be noted that this is the essential difference between the bounding technique in part (2) of the modified algorithm and that used in part (1). In the latter case the decoding is independent and we were able to obtain a tight bound on the decoding effort. However in part (2), the decoding is dependent, and we were forced to upper bound the probability that there is a set of k(R) streams that need more than ℓ steps to decode.

Now define $R^L_{boot}(\gamma)$ as the supremem of rates for which (25) is satisfied. Since the average computation will be bounded for the three group bootstrap decoding algorithm if $R < R^L_{boot}(1)$, $R^L_{boot}(1)$ is a lower bound on the R_{comp} of this coding scheme.

We can evaluate $R^L_{boot}(\gamma)$ by computing the differences

$$\frac{1}{\gamma}\min_{N}\left[\max\left(E_{\underline{n}}(\sigma_1)\Big|_{\sigma_1=\gamma}, E_{\underline{n}}(\sigma_2)\Big|_{\sigma_2=\gamma}, E_{\underline{n}}(\sigma_3)\Big|_{\sigma_3=\gamma}\right)\right] - \frac{k}{\gamma}E_\infty\left(\frac{\gamma}{k}\right) \tag{26}$$

for $k = 3,4,\ldots$ until their value becomes negative, where $N = \{\underline{n} = (n_1, n_2, n_3) \mid n_1 + n_2 + n_3 = k\}$. If this takes place for $k = k^+$, then

$$R^L_{boot}(\gamma) = \min\left\{\frac{1}{\gamma}\min_{N}\left[\max\left(E_{\underline{n}}(\sigma_1)\Big|_{\sigma_1=\gamma}, E_{\underline{n}}(\sigma_2)\Big|_{\sigma_2=\gamma}, E_{\underline{n}}(\sigma_3)\Big|_{\sigma_3=\gamma}\right)\right], \frac{k^+}{\gamma}E_\infty\left(\frac{\gamma}{k^+}\right)\right\} \tag{27}$$

where $N = \{\underline{n} = (n_1, n_2, n_3) \mid n_1 + n_2 + n_3 = k^+\}$.

DISCUSSION

In computing the lower bound on R_{comp} for the 3-group algorithm, we chose m, the number of streams, to be arbitrarily large since the lower bound on R_{comp} for the 1-group algorithm (single parity check case) was computed in the same way. However, minimizing over all N in this case always results in the situation in which 2 groups have been completely decoded, and all the remaining streams are in one group. Clearly, then, the lower bound on R_{comp} matches that for the 1-group case. If m is restricted to be rather small, say 21, some improvement in the bound over the 1-group case results, but not as much as could be expected for large m if the algorithm did not get into the worst case. Unfortunately, in the bound, we cannot restrict the choice of streams to decode in part (1), but in any real trial, it would be very unlikely that the worst case would result. Some computations were made under the assumption that the worst case could not occur, and significant improvement over the 1-group algorithm was noted.

REFERENCES

1. F. Jelinek and J. Cocke, "Bootstrap Hybrid Decoding for Symmetrical Binary Input Channels", Information and Control, Vol. 18, No. 3, April 1971.

2. F. Jelinek, "Study of Sequential Decoding", Final Report, NASA Contract NAS2-5643, 1972.

ADAPTIVE HYBRID SEQUENTIAL DECODING

L. BAHL, J. COCKE, F. JELINEK and J. RAVIV
IBM Research Center
Yorktown Heights, New York

ABSTRACT

In a recent paper, Jelinek and Cocke introduced a two-dimensional scheme which uses a long constraint length convolutional code in the longitudinal direction and a single parity check code in the transverse direction. We generalize the decoding method so that any linear code, block or convolutional, can be used in the transverse direction.

STRUCTURAL ANALYSIS OF CONVOLUTIONAL CODES VIA DUAL CODES

G. DAVID FORNEY, JR.[†]
Stanford Electronics Labs
Stanford, California 94305

ABSTRACT

A linear correspondence is developed between the states of a rate-k/n convolutional encoder G and the states of a corresponding syndrome former H^T, where H is an encoder of the code dual to the code generated by G. This correspondence is used to find an expression for the number of all-zero paths of length τ in the code trellis; the answer depends only on the constraint lengths of the dual code. A partial answer to the resynchronization problem also falls out of this development.

I. INTRODUCTION

In an earlier paper [1] we developed an arsenal of algebraic tools useful in the analysis of convolutional codes. In this paper we extend and apply some of these tools to attack some peripheral structural questions concerning convolutional codes.

The principal motivation for this work was the zero-run problem, for which partial results have recently been developed by a number of authors [2]-[4]. The object here is to determine the maximum number of consecutive all-zero branches that can occur in a non-zero code word; or, more generally, to determine the number of all-zero path segments of length τ branches in the code trellis, which is also the multiplicity of all distinct path segments of length τ since such segments form a vector space.

The solution to this problem can be used to provide a crude lower bound to the rate of increase of weight in non-zero code words, and as we shall see also gives a partial answer to the related question of resynchronization. However, in our view the interest of the zero-run problem lies mainly in the intriguing form of the solution. It turns out that the solution is stated in terms of certain parameters of the code that is dual to the code whose structure is being investigated, just as in the theory of block codes many results are most simply derived in terms of the code's parity-check matrix (the generator matrix of the dual code).

Consequently we are motivated to take a closer look at dual codes than our cursory survey in [1, Appendix I]. This paper develops some initial results on the analysis of convolutional codes via their dual codes; however, it seems likely that we have not exhausted the productiveness of this viewpoint, and that it may eventually be useful in exploring the *terra incognita* of weight and distance properties of convolutional codes.

Section II below reviews the results, notation, and viewpoint of [1]. Section III develops a linear correspondence (isomorphism) between the state spaces of a minimal encoder and the corresponding minimal syndrome former (parity checker), which correspondence has an interesting dynamical interpretation. The syndrome former is seen to contain sufficient information to specify the encoder state and inputs at any time; in particular it can be augmented without increase of memory to be an inverter. Section IV uses this dynamical correspondence to solve the zero-run problem, while in passing developing a potentially useful characterization of the paths preceding the following a given state in the code trellis.

[†] This work was supported by the National Science Foundation under Grant GK-31627. Part of this paper was presented at the 1972 IEEE International Symposium on Information Theory, Asilomar, California, February 1, 1972.

II. REVIEW OF [1]

We first review some of the principal concepts and results of [1], to which the reader may wish to refer for a more extensive discussion.

A rate-k/n convolutional encoder G is a k-input, n-output, time-invariant, invertible, realizable linear sequential circuit. In algebraic terms G is represented by a rank-k kxn matrix whose elements g_{ij} are causal rational functions; i.e., sequences (impulse responses) that start at time zero or later and can be expressed as ratios of polynomials in the delay operator D (D-transforms) or alternately in the indeterminate $z = D^{-1}$ (z-transforms). The sequence coefficients lie in some field F, which for coding is always finite, typically the binary field GF(2). The ring of polynomials in D over F is written conventionally F[D], and the field of rational functions F(D).

The convolutional code $\mathcal{C}$ generated by G is the set of all n-tuples of sequences $\underline{y} = \underline{x}G$, where $\underline{x}$ ranges through all k-tuples of sequences. The n-tuple $\underline{y}$ is called a codeword. By a sequence is meant the D-transform

$$x = x_d D^d + x_{d+1} D^{d+1} + \cdots \tag{1}$$

of a sequence $\{x_d, x_{d+1}, \ldots\}$ of elements of F that starts at some time d, possibly negative, and may continue forever. In algebra such sequences are sometimes called the field F((D)) of formal Laurent series in D over F (the field of quotients of the ring F[[D]] of formal power series). $\mathcal{C}$ is thus a vector space of dimension k over F((D)). (The rational codewords $\underline{y}$ are a subset $\mathcal{C}_r$ of the whole code $\mathcal{C}$, and form a vector space of dimension k over F(D).)

A realization (A,B,C,E) of an encoder G is a set of matrices over F of appropriate dimension such that the recursive equations

$$\begin{aligned} \underline{\sigma}_{t+1} &= \underline{\sigma}_t A + \underline{x}_t B \\ \underline{y}_t &= \underline{\sigma}_t C + \underline{x}_t E \end{aligned} \tag{2}$$

yield the transfer function matrix

$$G = E + B(I-AD)^{-1}CD. \tag{3}$$

Here $\underline{\sigma}_t$ is a vector over F, called the physical state, whose length is $\dim \Sigma_{(A,B,C,E)}$, the dimension of the physical state space $\Sigma_{(A,B,C,E)}$. The physical state $\underline{\sigma}_t$ is interpreted as the contents of $\dim \Sigma_{(A,B,C,E)}$ memory element at time t.

Other notions of state occur in linear system theory and in [1]. Let P be the projection operator that truncates sequences to end at time zero, and Q = 1-P the projection operator that truncates sequences to start at time 1; i.e.,

$$xP = \begin{cases} x_d D^d + \cdots + x_0, & d \leq 0; \\ 0, & d > 0 \end{cases}$$

$$xQ = x_1 D + x_2 D^2 + \cdots \tag{4}$$

Let $\underline{x}$ be any k-tuple of inputs; then the abstract state (at time 1) associated with $\underline{x}$ is $\underline{s}(\underline{x}) \triangleq \underline{x}PGQ$; i.e., the n-tuple of outputs at time 1 and later that are due to that part of $\underline{x}$ that occurs up to time 0. The abstract state depends only on G and not on the realization. The abstract state space Σ_G is the set of all abstract states; since P,G, and Q are linear over F, it is a linear vector space over F. Each physical state

$\underline{\sigma}$ leads to a definite abstract state

$$\underline{s} = \underline{\sigma}(I-AD)^{-1}CD; \qquad (5)$$

therefore $\dim \Sigma_{(A,B,C,E)} \geq \dim \Sigma_G$, with equality if and only if the map (5) is one-to-one (the realization is controllable and observable), when the realization is called irreducible. For irreducible encoders (5) exhibits an isomorphism between $\Sigma_{(A,B,C,E)}$ and Σ_G.

When G is a polynomial matrix, then it has a so-called obvious realization [1] in terms of shift registers, which in linear systems theory is called the controller canonical form [5]. Let us define the *i^{th} constraint length* as $\nu_i = \max_{1 \leq j \leq n} \deg g_{ij}$, $1 \leq i \leq k$; the ν_i are elsewhere called minimal (controllability) indices. The obvious realization involves k shift registers of lengths ν_i, $1 \leq i \leq k$, whose contents are the physical state $\underline{\sigma}_t = [x_{i,t-j}, 1 \leq i \leq k, 1 \leq j \leq \nu_i]$. The *overall constraint length* $\nu \triangleq \Sigma\nu_i$ is the dimension of the physical state space. Alternately $\underline{\sigma}_t$ may be represented by a k-tuple $\underline{\psi}$ of polynomials ψ_i in $D^{-1}=z$, where

$$\psi_i = x_{i,t-1} + x_{i,t-2}D^{-1} + \dots + x_{i,t-\nu_i} D^{-(\nu_i-1)} \qquad (6a)$$

and hence $\deg \psi_i \leq \nu_i - 1$.

For example, Fig. 1 illustrates the obvious realization of the rate-2/3 encoder

$$G = \begin{bmatrix} 1 & D & 1+D \\ D^2 & 1+D+D^2 & 1 \end{bmatrix} \qquad (7)$$

whose overall constraint length is 3. The physical state of this encoder (at time 1) in response to the k input sequences $\underline{x}$ may be written as

$$\underline{\psi} = \underline{x}P \bmod \underline{\beta} \qquad (6b)$$

where P is again the projection operator that truncates sequences to end at time 0, and $\underline{\beta}$ is the vector $[D^{-1}, D^{-2}]$ of polynomials in D^{-1}. In this case the obvious realization is irreducible; the isomorphism between physical states and abstract states is tabulated in Table I.

Two encoders are regarded as equivalent if they generate the same code. The main result of [1] is that for every code in the equivalence class of encoders that generate the code, there exists a so-called *minimal encoder* G, among whose properties are the following:

a) G is polynomial;

b) the obvious realization (controller canonical form) of G is irreducible;

c) every encoder G' generating $\mathcal{C}$ has $\dim \Sigma_{G'} \geq \dim \Sigma_G = \nu$. The last property is a consequence of the following technical lemma:

d) if $\underline{s}$ is a nonzero abstract state in Σ_G, then $\underline{s}$ is not a codeword in $\mathcal{C}$.

Minimal encoders are not in general unique, but all minimal encoders generating the same code have the same set of constraint lengths ν_i.

The dual $\mathcal{C}^{\perp}$ to a convolutional code $\mathcal{C}$ is the set of all n-tuples of sequences $\underline{z}$ such that the inner product $(\underline{y},\underline{z}) \triangleq \underline{y}\underline{z}^T$ (where T means transpose) is zero for all $\underline{y}$ in $\mathcal{C}$. The dual code $\mathcal{C}^{\perp}$ is a vector space of dimension n-k over $F((D))$, and may be generated by any rate-(n-k)/n encoder H such that $GH^T = 0$. The n-input, (n-k)-output linear sequential circuit whose transfer function matrix is H^T is called a syndrome former, and has the property that $\underline{y}H^T = 0$ if and only if $\underline{y} \in \mathcal{C}$.

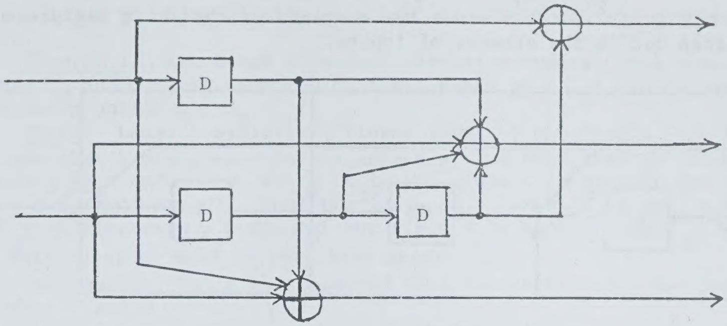


Fig. 1. Obvious Realization (Controller Canonical Form) of Encoder G in (7).

TABLE I

Physical state $\underline{\psi}$	Abstract state $\underline{s} = \underline{\psi}GQ$
$[0,\ 0]$	$[0,\ 0,\ 0]$
$[1,\ 0]$	$[0,\ D,\ D]$
$[0,\ 1]$	$[D^2,\ D+D^2,\ 0]$
$[1,\ 1]$	$[D^2,\ D^2,\ D]$
$[0,\ D^{-1}]$	$[D,\ D,\ 0]$
$[1,\ D^{-1}]$	$[D,\ 0,\ D]$
$[0,\ 1+D^{-1}]$	$[D+D^2,\ D^2,\ 0]$
$[1,\ 1+D^{-1}]$	$[D+D^2,\ D+D^2,\ D]$

Let H be a minimal encoder generating $\mathcal{C}^{\perp}$, and let its constraint lengths be $\nu_i^{\perp}$, $1 \leq i \leq n-k$. It was shown in [1] that the overall constraint lengths $\Sigma\nu_i$ and $\Sigma\nu_i^{\perp}$ of dual minimal encoders G and H are equal. Now if (A,B,C,E) is a realization of H, then (A^T, C^T, B^T, E^T) is a realization of H^T involving the same size state space; so if H is a minimal encoder, the corresponding syndrome former H^T can be realized in what we shall call an adjoint-obvious realization (observer canonical form), involving n-k shift registers of length $\nu_i^{\perp}$, $1 \leq i \leq n-k$. (Here the $\nu_i^{\perp}$ are the minimal observability indices.) Fig. 2 illustrates an adjoint-obvious realization for the syndrome former H^T, where

$$H = \left[\, 1+D+D^3,\ \ 1+D^2+D^3,\ \ 1+D+D^2+D^3 \,\right] \qquad (8)$$

is the minimal encoder dual to our earlier example encoder G. Note that in an adjoint-obvious realization the physical state is given in terms of the abstract state $\underline{s}^{\perp}$ by $\underline{\sigma} = [s_{ij}^{\perp},\ 1 \leq i \leq n-k,\ 1 \leq j \leq \nu_i^{\perp}]$. For example, in Figure 2 the physical state $\underline{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$ corresponds to the abstract state $\underline{s}^{\perp} = \sigma_1 D + \sigma_2 D^2 + \sigma_3 D^3$, since the contents of the shift registers are simply shifted out in the absence of inputs.

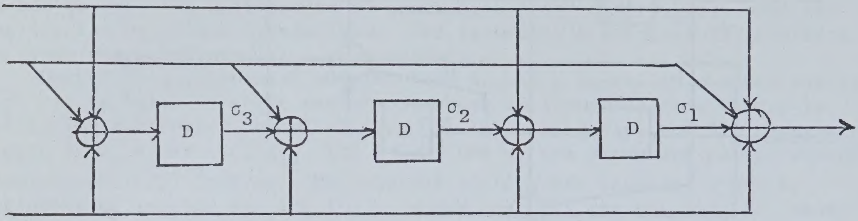


Fig. 2. Adjoint-Obvious Realization (Observer Canonical Form) of Syndrome Former H^T in (8).

TABLE II

Abstract state $\underline{s}$	Dual state $\underline{s}^{\perp} = \underline{s}H^T$
$[0, 0, 0]$	$[0]$
$[0, D, D]$	$[D^2]$
$[D^2, D+D^2, 0]$	$[D]$
$[D^2, D^2, D]$	$[D+D^2]$
$[D, D, 0]$	$[D^2+D^3]$
$[D, 0, D]$	$[D^3]$
$[D+D^2, D^2, 0]$	$[D+D^2+D^3]$
$[D+D^2, D+D^2, D]$	$[D+D^3]$

III. DUALITY BETWEEN SYNDROME FORMER AND ENCODER STATES

When the n encoder outputs $\underline{y} = \underline{x}G$ are connected directly to the syndrome-former inputs, the output of the syndrome former is zero for all time. In this section we observe that under these conditions the syndrome-former states mirror the encoder states, so that the syndrome former can be used as a state-tracker as well. This leads to the idea of labelling the encoder states by the syndrome-former states to which they correspond.

Lemma 1: Let G and H be dual encoders, and let the outputs of G drive the syndrome former H^T. Then whenever the abstract state of G is $\underline{s}$, the abstract state of H^T will be $\underline{s}^{\perp} = -\underline{s}H^T$.

Proof: Let $\underline{x}$ be any input that generates $\underline{s}$ ($\underline{s} = \underline{x}PGQ$), and let $\underline{y} = \underline{x}PG$, so that $\underline{s} = \underline{y}Q$. We have $\underline{y}H^T = 0$ since $GH^T = 0$; but

$$\begin{aligned} \underline{y}H^TQ &= \underline{y}PH^TQ + \underline{y}QH^TQ \\ &= \underline{s}^{\perp} + \underline{s}H^T, \end{aligned} \tag{9}$$

where $\underline{s}^{\perp} = \underline{y}PH^TQ$ is the syndrome-former state (at time 1) corresponding to the input $\underline{y}$. Consequently $\underline{s}^{\perp} = -\underline{s}H^T$ at time 1 and, by constancy, at all times.

Theorem 1: Let G and H be dual minimal encoders; then their state spaces Σ_G and Σ_{H^T} are isomorphic; i.e., every $\underline{s}^{\perp} \in \Sigma_{H^T}$ can be expressed as $-\underline{s}H^T$ for a unique $\underline{s} \in \Sigma_G$.

Proof: Lemma 1 exhibits a linear (over F) map from $\underline{s} \in \Sigma_G$ to $\underline{s}^{\perp} \in \Sigma_{H^T}$. To show that every $\underline{s}$ maps into a unique $\underline{s}^{\perp}$, we note that $\underline{s}H^T$ is not zero unless $\underline{s}$ is a code word, which it is not since G is minimal and thus $\underline{s} \neq 0 \Longrightarrow \underline{s} \notin \mathcal{C}$ (property d). Thus the image of Σ_G under this map is a subspace of Σ_{H^T} of dimension $\nu = \dim \Sigma_G$; but since H is minimal, $\dim \Sigma_{H^T} = \dim \Sigma_G$, and this subspace must be the whole space.

For example, Table II tabulates this isomorphism for our example encoders G and H.

Lemma 1 and Theorem 1 suggest another label for an encoder state, namely the dual state $\underline{s}^{\perp} = -\underline{s}H^T$. Thus Table II may be regarded as an extension of Table I, and gives another dynamics-preserving isomorphism of the state space.

The fact that a syndrome former H^T is a faithful Doppelganger shadowing the state sequence of an encoder G implies that by adding to H^T combinational logic realizing the matrix over F implied by the isomorphism and ν additional outputs we may add the current encoder state as another output of H^T. This can be any representation of the encoder state, such as $\underline{\psi}$, $\underline{\sigma}$ or $\underline{s}$; the dual state $\underline{s}^{\perp}$ can of course be read out directly. But this implies that with a little more combinational logic and k more outputs we can read out the current encoder input $\underline{x}_t$ as well; for the encoder output equation

$$\underline{y}_t = \underline{\sigma}_t C + \underline{x}_t E \tag{10}$$

can be inverted to give

$$\underline{x}_t = \underline{y}_t E^{-1} - \underline{\sigma}_t C E^{-1} \tag{11}$$

where we know E has a right inverse E^{-1} since G is minimal (see [1]; otherwise there would be an anticausal input giving a causal output). Hence we can derive an inverter G^{-1} without the addition of any memory elements. These ideas are illustrated in Figure 3 for a general encoder syndrome former pair and in Figure 4 for our example G and H.

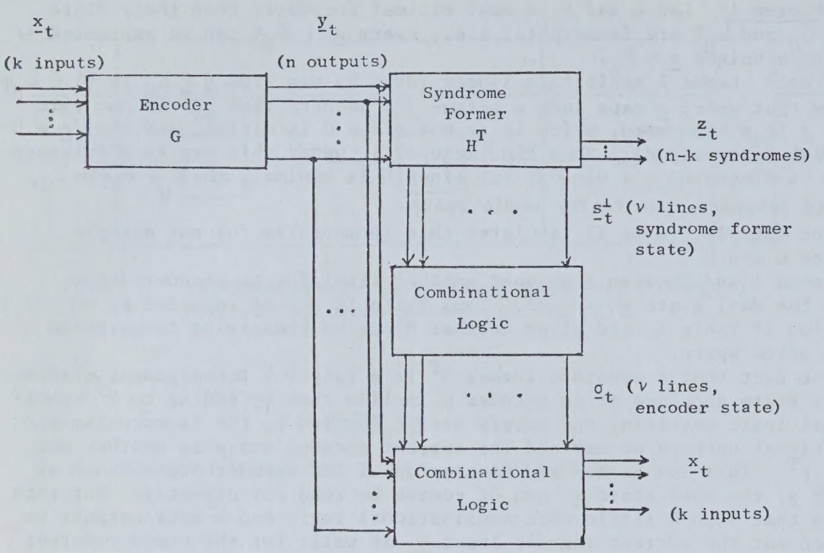
Moore and Silverman [6] have recently shown that any circuit G with a feedforward inverse has an inverse that can be realized with no more than $\dim \Sigma_G$ memory elements. The above argument provides a rather different proof of this result when G is minimal. It should be noted that much simpler inverses are usually obtainable.

IV. THE ZERO-RUN PROBLEM

A number of authors [2]-[4] have recently investigated the problem of zero runs in convolutional codes. If an encoder G does not start in the zero state, for how many consecutive time units τ can its outputs $\underline{y}_t$, $1 \leq t \leq \tau$, be all zero?

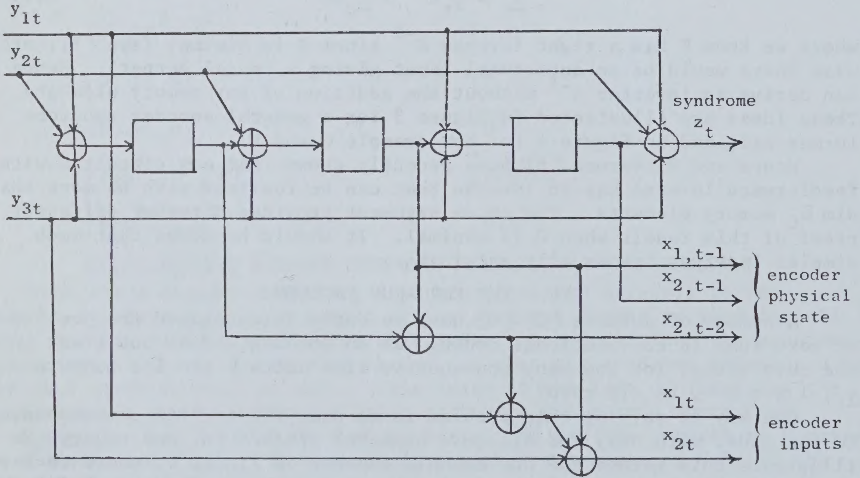
One way of solving this problem is to draw the trellis corresponding to the code, with only the all-zero branches blacked in, and count. We illustrate this method for our example encoder in Figure 5, where we have used both the conventional physical state $\underline{\psi}$ and the dual state $\underline{s}^{\perp}$ introduced in the last section as labels. We see that there are 4 all-zero branches, two of which combine to form a single all-zero path of length 2 unmerged with the zero state. It is suggestive that the transitions in $\underline{s}^{\perp}$ corresponding to this length-2 path are simply $D^3 \rightarrow D^2 \rightarrow D$.

We now give the general solution to this problem. We develop the result through rather physical arguments.



$\underline{x}_t$
(k inputs)
$\underline{y}_t$
(n outputs)
Encoder
G
Syndrome Former
H^T
$\underline{z}_t$
(n-k syndromes)
$\underline{s}^{\perp}_t$ (ν lines, syndrome former state)
Combinational Logic
$\underline{\sigma}_t$ (ν lines, encoder state)
Combinational Logic
$\underline{x}_t$
(k inputs)

Fig. 3. Tracking of Encoder State and Inputs from Syndrome-Former State and Inputs



y_{1t}
y_{2t}
y_{3t}
syndrome
z_t
$x_{1,t-1}$
$x_{2,t-1}$
$x_{2,t-2}$
encoder physical state
x_{1t}
x_{2t}
encoder inputs

Fig. 4. Tracking of Encoder State and Inputs for Example Encoder and Syndrome Former

Let $\nu^{\perp}_{\max} = \max \nu^{\perp}_{i}$; let $[H^T]_j$, $0 \leq j \leq \nu^{\perp}_{\max}$, be the matrices of coefficients of D^j in H^T; and let $[\underline{s}^{\perp}_{t}]_j$, $1 \leq j \leq \nu^{\perp}_{\max}$, be the coefficients D^j in the syndrome-former abstract state $\underline{s}^{\perp}_{t}$ at time t. In the adjoint-obvious realization, the reader can verify that the state update and output equations have the shift-register form

$$\begin{aligned} [\underline{s}^{\perp}_{t+1}]_j &= [\underline{s}^{\perp}_{t}]_{j+1} + \underline{y}_t[H^T]_j; \\ \underline{z}_t &= [\underline{s}^{\perp}_{t}]_1 + \underline{y}_t[H^T]_0 \end{aligned} \tag{12}$$

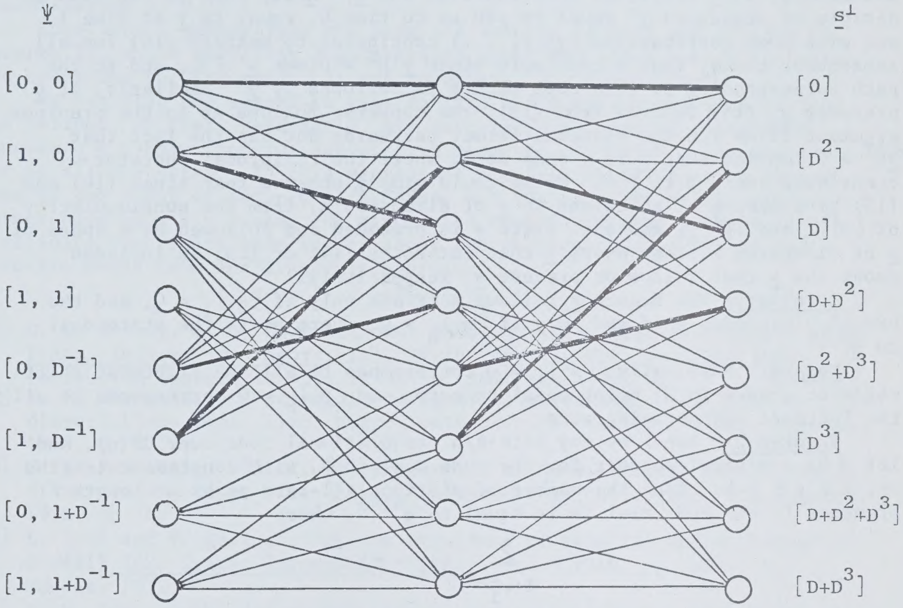


Fig. 5. Code Trellis for Example Code
with All-Zero Branches Darkened

Furthermore, let $[H^T]_h$ be the matrix of highest-order coefficients in H^T -- that is, the ith column of $[H^T]_h$ consists of the coefficients of $D^{\nu^{\perp}_{i}}$ in the ith column of H^T --and let $[\underline{s}^{\perp}_{t}]_h$ be the highest-order coefficients in $\underline{s}^{\perp}_{t}$, i.e., the coefficients of $D^{\nu^{\perp}_{i}}$, or the contents of the leftmost shift-register stages. Then since these stages have no inputs from previous stages, their update equation is simply

$$[\underline{s}^{\perp}_{t+1}]_h = \underline{y}_t[H^T]_h. \tag{13}$$

It was proved in [1] that when H is minimal both $[H^T]_0$ and $[H^T]_h$ have rank n-k.

In an encoder G, let there be an input k-tuple $\underline{x}$ (over F) that causes a transition from state $\underline{s}_1$ to state $\underline{s}_2$ and generates the output n-tuple $\underline{y}$ (over F), called a branch. Then we say the branch $\underline{y}$ follows $\underline{s}_1$ and precedes $\underline{s}_2$.

Lemma 2: The branch $\underline{y}$ follows $\underline{s}$ if and only if

$$\underline{y}[H^T]_0 = -[\underline{s}^\perp]_1 \tag{14}$$

and the branch $\underline{y}$ precedes $\underline{s}$ if and only if

$$\underline{y}[H^T]_h = [\underline{s}^\perp]_h \tag{15}$$

where $\underline{s}^\perp$ is the dual state corresponding to $\underline{s}$.

Proof: If $\underline{y}$ follows $\underline{s}$, then the syndrome-former output at that time is zero, so from (12)

$$0 = [\underline{s}^\perp]_1 + \underline{y}[H^T]_0. \tag{16}$$

Conversely, if $\underline{y}$ satisfies this equation and $\underline{s} = \underline{x}PGQ$, then we can find an n-tuple of sequences $\underline{y}'$ equal to $\underline{x}PG$ up to time 0, equal to $\underline{y}$ at time 1, and with some continuation $(\underline{y}_2', \underline{y}_3', \ldots)$ continuing to satisfy (16) for all subsequent times, thus a code word since $\underline{y}'H^T = 0 \Longrightarrow \underline{y}' \in \mathcal{C}$, and in the path corresponding to this code word $\underline{s}$ is followed by $\underline{y}$. Similarly, if $\underline{y}$ precedes $\underline{s}$, (15) is true from (13); the converse follows as in the previous argument if we run the syndrome-former backwards and use the fact that $\tilde{\underline{y}}H^T = 0$ implies that $\underline{y}$ is a code word, where the reciprocal operator $\sim$ transforms every D to D^{-1}. Or we could simply observe that since (14) and (15) have spaces of solutions in $\underline{y}$ of dimension k, from the nonsingularity of $[H^T]_0$ and $[H^T]_h$, and each state $\underline{s}$ is preceded and followed by a space of $\underline{y}$ of dimension k, then every $\underline{y}$ that satisfies (14) or (15) is included among the $\underline{y}$ that follow or precede $\underline{s}$, respectively.

Corollary: The branch $\underline{0}$ follows $\underline{s}$ if and only if $[\underline{s}^\perp]_1 = \underline{0}$, and the branch $\underline{0}$ precedes $\underline{s}$ if and only if $[\underline{s}^\perp]_h = \underline{0}$, where $\underline{s}^\perp$ is the state dual to $\underline{s}$.

Remark: Physically, $[\underline{s}^\perp]_1 = \underline{0}$ corresponds to all the rightmost shift-register stages in H^T being equal to zero, and $[\underline{s}^\perp]_h = \underline{0}$ corresponds to all the leftmost stages being zero.

Theorem 2: Let $\mathcal{C}$ be any rate-k/n convolutional code over GF(q), and let H be a minimal encoder for the code dual to $\mathcal{C}$, with constraint lengths $\nu_i^\perp$, $1 \leq i \leq n-k$. Then the number of distinct all-zero paths of length τ branches in the code trellis is equal to $q^{d(\tau)}$, where

$$d(\tau) = \sum_{i:\nu_i^\perp \geq \tau} (\nu_i^\perp - \tau) \tag{17}$$

Proof: Let $\underline{s}$ be any state in the trellis, let $\underline{s}^\perp$ be the corresponding dual state, and let $\tau(\underline{s})$ be the minimum j for which $[\underline{s}^\perp]_{j+1} \neq 0$. By the corollary above, if $\tau(\underline{s}) > 0$ then there is exactly one all=zero path of length $\tau(\underline{s})$ starting at $\underline{s}$, namely the path identified by the dual state sequence

$$\underline{s}^\perp \rightarrow D^{-1}\underline{s}^\perp \rightarrow \ldots \rightarrow D^{-\tau(\underline{s})}\underline{s}^\perp. \tag{18}$$

Thus the number of paths of length τ in the trellis is precisely the number of states whose dual states have their τ lowest-order coefficients equal to zero (or physically, for which the τ rightmost stages in all shift registers are empty). The number of degrees of freedom in specifying these states is the number of higher-order coefficients (the remaining shift-register stages), and is given by $d(\tau)$.

Corollary 1: Let $\mathcal{C}$ be any rate-(n-1)/n convolutional code over GF(q) whose minimal encoder has overall constraint length ν. Then the number of all-zero paths of length τ in the code trellis is $q^{\nu-\tau}$ for $\nu \geq \tau$ and 1 otherwise (the infinite all-zero path).

Proof: Here n-k = 1 and $\nu_1^\perp = \nu$ since $\dim \Sigma_{H^T} = \dim \Sigma_G$.

Corollary 2: Let $\nu_{\max}^\perp = \max \nu_i^\perp$; then under the conditions of Theorem 2 the longest all-zero path nowhere merged with the zero state has length

$\nu^{\perp}_{\max} - 1$. The quantity $\nu^{\perp}_{\max}$ is bounded by

$$\left\lceil \frac{\nu}{n-k} \right\rceil \leq \nu^{\perp}_{\max} \leq \nu, \tag{19}$$

where $\lceil x \rceil$ is the least integer not less than x, and ν is the over-all constraint length of any minimal encoder generating $\mathcal{C}$.

Proof: $\sum_{i=1}^{n-k} \nu^{\perp}_i = \nu$.

Remark: Either bound in (19) can be achieved with equality by proper specification of H.

Corollary 1 generalizes the rate-1/2 results of Bahl and Jelinek [4, Theorem 2], while Corollary 2 generalizes upper and lower bounds for rate-1/n codes due to Odenwalder [2] and Rosenberg [3], respectively.

We can also now extend to rate-(n-1)/n codes the observation of Bahl and Jelinek [4] that, remarkably, all rate-(n-1)/n codes of constraint lengths $\nu \geq \tau$ have the same set of path segments of length τ, namely the set of all $n\tau$-tuples. In this sense, the simplest class of codes to analyze is the class of rate-(n-1)/n codes, not the rate-1/n codes, since the former have rate-1/n duals. We might also remark that the present results invalidate the remark made in [1] about the absence of new results for rate-1/n codes from our algebraic approach.

REFERENCES

[1] G. D. Forney, Jr., "Convolutional Codes I: Algebraic Structure," IEEE Trans. Information Theory, Vol. IT-16, pp. 720-738, November 1970; correction appears in Vol. IT-17, p. 360, May 1971.

[2] J. P. Odenwalder, "Optimal Decoding of Convolutional Codes," Ph.D. dissertation, Dept. Elec. Engr., University of California, Los Angeles, 1970, p. 47.

[3] W. J. Rosenberg, "Structural Properties of Convolutional Codes," Ph.D. dissertation, Dept. Elec. Engr., University of California, Los Angeles, 1971, Ch. 4.

[4] L. Bahl and F. Jelinek, "On the Structure of Rate 1/n Convolutional Codes," IEEE Trans. Information Theory, Vol. IT-18, pp. 192-196, January 1972.

[5] H. H. Rosenbrock, State-space and Multivariable Theory. New York: Wiley, 1970.

[6] B. C. Moore and L. M. Silverman, "A New Characterization of Feed-Forward, Delay-Free Inverses," submitted to IEEE Trans. Information Theory.

[7] G. Basile and G. Marro, "On the Observability of Linear Time-Invariant Systems with Unknown Inputs," J. Optimization Theory Appl. Vol. 3, No. 6, 1969.

[8] D. Rappaport and L. M. Silverman, "Structure and Stability of Discrete-Time Optimal Systems," IEEE Trans. Automatic Control, Vol. AC-16, pp. 227-233, June 1971.

[9] E. A. Bucher and J. A. Heller, "Error Probability Bounds for Systematic Convolutional Codes," IEEE Trans. Information Theory, Vol. IT-16, pp. 219-224, March 1970.

PROPERTIES OF LINEAR TIME-INVARIANT MULTIVARIABLE SYSTEMS SUBJECT TO ARBITRARY OUTPUT AND STATE FEEDBACK

E. J. DAVISON* AND S. H. WANG*

University of Toronto
Toronto 181, Canada

ABSTRACT

It is shown that for any linear time-invariant multivariable system which is both completely controllable and completely observable, almost all output feedback laws can be used to make the closed loop system have distinct poles; i.e., the set of output feedback laws which fails to achieve this goal is either an empty set or a hypersurface in the parameter space. It is also shown that almost any output feedback law will make the closed loop system's poles be disjoint from any given finite set of points on the complex plane. It is then shown that any controllable observable system can be made controllable and observable with respect to any nontrivial input and output by applying almost all output feedback laws about the original system. These results have immediate application in pole assignment, in the controllability of parallel connected systems and in the identification problem. In addition, it is shown that for any linear system, for almost all state feedback laws, the closed loop system has the property that all modes of the system except possibly those corresponding to the uncontrollable-unobservable part of the system are observed in the output.

I. INTRODUCTION

This paper deals with the properties of multivariable linear systems subject to arbitrary output or state feedback, i.e., if an arbitrary feedback law is chosen, then what can be said about the resultant controlled system? Besides being interesting in itself, the motivation for this topic also arises in sensitivity analysis, i.e., it is an obvious fact that in practise, it is impossible to implement a feedback configuration $u = K^*x$, say, using precisely the gains K^* -

---

* Department of Electrical Engineering, University of Toronto, Toronto, Canada. This work has been supported by the National Research Council of Canada under Grant No. A4396.

there is always some error δK^* in K^* which is not known a priori, and so one can think of the error as corresponding to a feedback control $u = \delta K^* x$ where δK^* is arbitrary but small. In the following, only the statement of the main theorems will be given and all proofs will be omitted.

II. MATHEMATICAL PRELIMINARIES

Let $\mathbb{R}$ and $\mathbb{C}$ denote the fields of real and complex numbers, respectively. The ring of polynomials in n variables over $\mathbb{R}$ is written $\mathbb{R}[x_1,\ldots,x_n]$. If $f = f(x_1,\ldots,x_n)$ is a polynomial in the n variables $x_1,\ldots,x_n$ with coefficients in $\mathbb{R}$, then the point $\hat{x} = (\hat{x}_1,\ldots,\hat{x}_n)$ in $\mathbb{R}^n$ is said to be a zero of f if $f(\hat{x}_1,\ldots,\hat{x}_n) = 0$. The set of zeros of f is called the locus of f, and is denoted by $V(f)$. A subset V of $\mathbb{R}^n$ is called a hypersurface in $\mathbb{R}^n$ if it is the locus of a non-constant polynomial. For detailed discussions and examples on a hypersurface, see [1], [2].

III. SEPARATION OF SYSTEM POLES

Consider a linear time-invariant multivariable system specified by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.a}$$

$$y(t) = Cx(t) \tag{1.b}$$

where x is an n-dimensional state vector, u is an m-dimensional input vector, y is a p-dimensional output vector and A, B and C are real constant matrices of appropriate dimensions. The following result is obtained:

Theorem 1

If the system in (1.a,b) is completely controllable and completely observable, then the set

$$\mathcal{K} = \{K \mid K \in \mathbb{R}^{m\times p} \text{ and } (A+BKC) \text{ has repeated eigenvalues}\} \tag{2}$$

is either an empty set or a hypersurface in $\mathbb{R}^{m\times p}$.

Remark 1

Since a matrix with no repeated eigenvalues implies that it is cyclic, then the following set:

$$\mathcal{K}^* = \{K \mid K \in \mathbb{R}^{m\times p} \text{ and } (A+BKC) \text{ is noncyclic}\} \tag{3}$$

is clearly a subset of K defined in (2). It can be shown that the set K^* is also a hypersurface in $\mathbb{R}^{m\times p}$.

From Theorem 1 it can therefore be seen that almost any output feedback law will make a system have distinct eigenvalues. The following algorithm can therefore be used to calculate such a feedback law and has immediate application in the pole assignment problem:

Algorithm to find Output Feedback Law to separate System's Poles

Step (1) Put $i = 1$, $\tilde{A} = A$, $K = 0$

Step (2) Choose an arbitrary matrix K_i, by using a pseudo-random number generator to generate the elements of K_i, so that $\|A\| \doteq \|BK_iC\|$ (This can always be done by properly scaling the output of the pseudo-random number generator). If the matrix $(\tilde{A}+BK_iC)$ has more distinct eigenvalues than the matrix $\tilde{A}$ then replace $\tilde{A}$ by $(A+BK_iC)$ and replace K by $K + K_i$, otherwise do not alter $\tilde{A}$ or K.

Step (3) Replace i by $i+1$ and repeat steps (2),(3) until the matrix A has all distinct eigenvalues. The final feedback matrix K is the desired feedback gain matrix.

IV. POLE-SHIFTING RESULT

In this section, it will be shown that for any given finite set of points on the complex plane $\mathbb{C}$, it is always possible to find an output feedback law so that the system's controllable and observable modes do not coincide with the given set. In fact almost all output feedback laws can do this.

Theorem 2

If the system in (1.a,b) is completely controllable and completely observable, then for any finite number of points $P \triangleq \{p_1,\ldots,p_j\}$ on the complex plane $\mathbb{C}$, the set

$$\tilde{K} = \{K \mid K \varepsilon \mathbb{R}^{m\times p} \text{ and } \sigma[A+BKC] \cap P \neq \phi\} \qquad (4)$$

is either an empty set or a hypersurface in $\mathbb{R}^{m\times p}$.

Remark 2 This theorem implies that given two controllable and observable linear systems connected in parallel, then the addition of almost any output feedback law about either system makes the resulting system controllable and observable.

V. CANONICAL STRUCTURE THEOREM FOR LINEAR SYSTEMS IN WHICH AN ARBITRARY OUTPUT FEEDBACK LAW HAS BEEN APPLIED

An interesting application of Theorems 1 and 2 can be made to obtain a simpler form of Kalman's Canonical Structure Theorem for a linear system in which output feedback is allowed. The following theorem is obtained:

Theorem 3 (Output Feedback Canonical Structure Theorem)

Given the system (1), assume that an arbitrary output feedback law has been applied; then for almost all output feedback laws, (1) can be decomposed into the following form:

$$\dot{x} = \begin{bmatrix} \tilde{A}_1 & 0 & 0 & 0 \\ 0 & \tilde{A}_2 & \tilde{A}_5 & \tilde{A}_6 \\ 0 & 0 & \tilde{A}_3 & \tilde{A}_7 \\ 0 & 0 & 0 & \tilde{A}_4 \end{bmatrix} x + \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \\ 0 \\ 0 \end{bmatrix} u \qquad (5)$$

$$y = (\tilde{C}_1 \quad 0 \quad 0 \quad \tilde{C}_2)\, x$$

where $\left\{ \begin{bmatrix} \tilde{A}_1 & 0 \\ 0 & \tilde{A}_2 \end{bmatrix}, \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} \right\}$ is controllable and $\left\{ (\tilde{C}_1, \tilde{C}_2), \begin{bmatrix} \tilde{A}_1 & 0 \\ 0 & \tilde{A}_4 \end{bmatrix} \right\}$ observable and where the eigenvalues of $\tilde{A}_1$ are all distinct, and where the dimensions of the matrices $\tilde{A}_1, \tilde{A}_2, \tilde{A}_3, \tilde{A}_4$ and the eigenvalues of $\tilde{A}_2, \tilde{A}_3, \tilde{A}_4$ are independent of the output feedback law applied.

Remark 3 If the arbitrary output feedback law chosen has "small enough gains", then the controlled system (1) will behave arbitrarily close to the original uncontrolled system (1) [3]. Since one can always apply an arbitrary output feedback law about the physical system described by (1) in practise, this means that one can always assume that Theorem 3 is applicable to any system described by (1) in which an arbitrary "small" output feedback law has been applied.

VI. OBSERVABILITY - CONTROLLABILITY RESULTS

In this section results on observability and controllability of systems under feedback control are given. In general it will be shown that observability in a system is preserved for almost all state feedback laws.

Consider a linear multivariable system given by (1), and assume that the pair (A,C) is completely observable but that (A,B) is not necessarily controllable, then the following result on observability can be made:

Proposition 1

Consider the completely observable system in (1). The set of state feedback gains F which makes the closed loop system not completely observably is either an empty set or a hypersurface in the parameter space of F.

A different type of result can be obtained if it is assumed that (A,B,C) of (1) is complete:

Theorem 4 Consider a linear multivariable system specified by the following equations,

$$\dot{x}(t) = Ax(t) + Bu(t) + gv(t) \tag{6}$$

$$y(t) = Cx(t) \tag{7}$$

$$z(t) = hx(t) \tag{8}$$

where $u(t) \varepsilon \mathbb{R}^m$ and $v(t) \varepsilon \mathbb{R}^1$ are inputs, $x(t) \varepsilon \mathbb{R}^n$ is the state, $y(t) \varepsilon \mathbb{R}^p$ and $z(t) \varepsilon \mathbb{R}^1$ are outputs, and A,B,C,g,h are real constant matrices and vectors of appropriate size. Assume that the triple (A,B,C) is complete, i.e., the pair (A,B) is completely controllable and the pair (A,C) is completely observable. Both g and h are assumed to be nonzero vectors. Then the following set

$$\mathcal{K} = \{K | K \varepsilon \mathbb{R}^{mxp} \text{, and } (A+BKC,g,h) \text{ is not complete}\} \tag{9}$$

is either an empty set or a hypersurface in the parameter space of K.

Remark 4 Theorem 1 and Theorem 4 should be very useful in the identification of linear multivariable systems; the addition of almost any output feedback law will make a controllable observable system cyclic and controllable and observable with respect to any nontrival input and output, thereby enabling a standard "canonical" structure to be assumed for the system [4]. It should be noted that in such an identification procedure, it is not necessary to actually implement an output feedback law; e.g., if u and y are the input and output data of a system to be identified, then if the data is processed so that $v \triangleq (u-Ky)$ and y are taken to be the new input and output data, where K is chosen at random, the new system with input v and output y may be assumed to be cyclic. This new system then identified as being (A,B,C) say of (1), and in this case the original system corresponds to (A-BKC,B,C).

Consider now the case of an arbitrary system (1) which may not be controllable or observable.

Definitions Let the Jordan Canonical form of the nxn matrix A be given by $A = \Gamma\Lambda\Gamma^{-1}$ where $\Lambda = \mathrm{diag}({}_1\Lambda, {}_2\Lambda, \ldots, {}_q\Lambda)$ where ${}_i\Lambda$ consists of all the Jordan blocks associated with the eigenvalue λ_i, $i=1,2,\ldots,q$, where λ_i are all distinct. Let these eigenvalues $\lambda_i, i=1,2,\ldots,q$, be called the modal eigenvalues of A or alternatively the modes of the system. The system $\dot{x}=Ax$, $y=Cx$ is said to be observable with respect to the modes $\lambda_r, \lambda_s, \ldots$ if there exists an initial state x(0) such that $y(t)=Ce^{At}x(0)$ only contains terms containing $e^{\lambda_r t} p_r(t)$, $e^{\lambda_s t} p_s(t), \ldots$, where $p_r(t), p_s(t), \ldots$ are nonzero polynomial vectors of t.

Consider now the case of a system (1) which may not be controllable or observable. The following result on observability can be made:

Theorem 5 Consider the following system which is not necessarily controllable or observable:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \qquad (10)$$

Let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ be the set of modes of the uncontrollable and observable subsystem of (10). Then the class of state feedback laws $u = Kx$ which makes any subset of Λ unobservable in y is either an empty set or a subset of a hypersurface in the space of K.

Remark 5 Theorem 5 is most interesting. In its proof we have shown that for any arbitrary triple (A,B,C) then for almost all K, the triple (A+BK,B,C) has the property that all modes of the system, except possibly those corresponding to the uncontrollable - unobservable part of the system, are observed in the output.

VII. SOME NUMERICAL EXAMPLES AND AN APPLICATION OF THE PREVIOUS RESULTS TO "ZEROING THE OUTPUT" PROBLEM

Given the controllable, observable non-cyclic triple (A,B,C) where

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (11)$$

it is desired to find a feedback control system u = Ky so that (A+BKC) is cyclic with distinct eigenvalues. In this case the following eigenvalues of (A+BKC) were found for different values of K, the K matrices being arbitrarily chosen:

(i) With $K = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ the eigenvalues of (A+BKC) are:

$(-1.72 \pm j1.17, -1.23, 1.18, 0.243 \pm j0.762)$,

(ii) with $K = \begin{bmatrix} 10 & -20 \\ 30 & 40 \end{bmatrix}$ the eigenvalues of (A+BKC) are:

$(-3.67 \pm j1.81, -0.466 \pm j3.40, 2.63 \pm j1.80)$,

(iii) with $K = \begin{bmatrix} -0.001 & -0.005 \\ 0.007 & 0.008 \end{bmatrix}$ the eigenvalues of (A+BKC) are:

$(-2.00, -1.01, 0.0329 \pm j0.0638, -0.0293 \pm j0.0659)$.

It can be seen that in all cases the eigenvalues of the resultant closed loop system matrix are distinct as predicted.

A Solution to "Zeroing the Output" Problem

Consider the following problem which is a modification of a problem considered by Bhattacharyya, Pearson and Wonham [5];

Given the linear constant system (1), which is not necessarily controllable or observable, it is desired to find conditions for there to exist a control $u=KTx$, where T is an unspecified coordinate transformation, so that $y(t)\to 0$ as $t\to\infty$ $\forall$ $x(0)\in\mathbb{R}^n$ with specified dynamic behaviour i.e., so that the characteristic exponents σ_i, $i=1,2,\ldots,\delta$, of $y(t)$ are contained in a preassigned symmetric subset $\mathbb{C}^-$ of the open left hand part of the complex plane $\mathbb{C}$ (in the sense that if a characteristic exponent $\lambda \in \mathbb{C}^-$ then its conjugate $\lambda^* \in \mathbb{C}^-$) and such that the control system obtained is robust for small perturbations of feedback gains in the system i.e., so that $y(t)\to 0$ as $t\to\infty$ with $\sigma_i \in \mathbb{C}^-$, $i=1,2,\ldots,\delta$, $\forall$ $x(0)\in\mathbb{R}^n$, $\forall$ $\delta K\in\Omega(\varepsilon)$, where $\delta K=\{\delta k_{ij}\}$ is a perturbation of $K=\{k_{ij}\}$ and $\Omega(\varepsilon)$ is defined as follows :

$$\Omega(\varepsilon) = \{\delta K \mid \delta K\in\mathbb{R}^{m\times n},\ |\delta k_{ij}|<\varepsilon, \text{ if } k_{ij}\neq 0, \text{ and } \delta k_{ij}=0 \text{ if } k_{ij}=0\},$$

where $\varepsilon > 0$. (12)

Definitions Let $\mathbb{C} = \mathbb{C}^- \cup \mathbb{C}^+$, $\mathbb{C}^- \cap \mathbb{C}^+ = \phi$ be a partition of $\mathbb{C}$ into symmetric subsets. Let $\sigma(A)$ denote the eigenvalues of A.

The following solution is given:

Theorem 6

Given the triple (A,B,C) defined by (1) and a symmetric subset $\mathbb{C}^-$ of $\mathbb{C}$, then a necessary and sufficient condition that there exists a control law $u=KTx$ and an $\varepsilon>0$ such that $\forall$ $x(0)\in\mathbb{R}^n$, $\forall$ $\delta K\in\Omega(\varepsilon)$ so that $y(t)\to 0$ as $t\to\infty$ with $\sigma_i\in\mathbb{C}^-$, $i=1,2,\ldots,\delta$, is that

$$\sigma(\tilde{A}_4) \subset \mathbb{C}^-, \tag{13}$$

where $\tilde{A}_4$ is given by the Canonical Form of (A,B,C) in (5).

Remark 6 In this problem formulation, it is assumed that an arbitrary but exact coordinate transformation $z=Tx$ can be made, and that on applying the control law $u=KTx$ only the nonzero elements of K are perturbed.

VIII. CONCLUSIONS

In this paper, it has been shown that for any linear time-invariant multivariable system, which is both completely controllable and completely observable, almost all output feedback laws can be used to make the closed loop system have distinct poles. This result should be very useful in the design of linear multivariable systems in which there often is an advantage in assuming that the system matrix is cyclic e.g., in pole assignment for multivariable systems. Theorem 2 shows that almost any output feedback law will make the closed loop system's poles be disjoint from any given set of points and Theorem 3 gives an "Output Feedback Canonical Structure Theorem". Theorem 4 shows that any controllable-observable linear system can be made controllable and observable with respect to any nontrivial input and output by applying almost any output feedback about the original system. This theorem has application in the identification problem. Theorem 5 shows that for any arbitrary triple (A,B,C) then for almost all K, the triple (A+BK,B,C) has the property that all modes of the system, except possibly those corresponding to the uncontrollable-unobservable part of the system, are observed in the output. As an application of how Theorem 5 may be useful in sensitivity problems, Theorem 6 deals with "zeroing the output" problem; it shows that the only way a system output can be driven to zero by applying state feedback and allowing arbitrary small perturbations about the feedback gains, is that the uncontrollable-observable modes of the system be stable.

REFERENCES

[1] W. Fulton, Algebraic Curves, W. A. Benjamin, Inc. 1969.

[2] J. Fogarty, Invariant Theory, W. A. Benjamin, Inc. 1969.

[3] E. A. Coddington and N. Levinson, Theory of Ordinary Differential Equations, McGraw Hill, 1955, p.58.

[4] K. J. Åström and P. Eykhoff, "System Identification - A Survey" Automatica, Vol. 7, 1971, pp. 123-162 (see pp. 133-134).

[5] S. P. Bhattacharyya, J. B. Pearson and W. M. Wonham, "On Zeroing the Output of a Linear System", Information and Control, Vol. 20, No. 2, March 1972, pp. 135-142.

L^2-STABILITY OF DISTRIBUTED FEEDBACK SYSTEMS: SINGULAR PERTURBATION

J. F. BARMAN, F. M. CALLIER[†] AND C. A. DESOER
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

We consider a continuous time, single input-single output, linear, time-invariant, distributed feedback system F^ε containing a small delay of length ε in the loop. Conditions are given under which L^2-stability and L^2-instability of this feedback system can be deduced from those of the reduced model obtained by neglecting the delay.

The two system models associated with F^ε are the unperturbed model F^0 and the high frequency model F_H^ε. The condition for neglecting the small delay is the L^2-stability of the family of high-frequency models $(F_H^\varepsilon)_{\varepsilon\geq 0}$, where $\varepsilon \geq 0$ is sufficiently small.

The paper contains a lemma and a theorem. The lemma gives sharp Nyquist-type conditions for the L^2-stability and L^2-instability of the family of high-frequency models $(F_H^\varepsilon)_{\varepsilon\geq 0}$ for sufficiently small $\varepsilon \geq 0$, while the Theorem gives explicit conditions under which the small delay may or may not be neglected.

Description of the system model, unperturbed and high-frequency models

We consider a family of continuous-time, scalar linear time-invariant feedback systems $(F^\varepsilon)_{\varepsilon\geq 0}$ with input u, error e and output y. u, e and y are functions mapping $\mathbb{R}_+$ into $\mathbb{R}$ and satisfy

$$y = T_\varepsilon(g * e) \tag{1}$$

$$e = u - y \tag{2}$$

where, * denotes the convolution operator, T_ε is a delay of length $\varepsilon \geq 0$; g is a real-valued distribution with support on $\mathbb{R}_+$.

Let $\hat{g}$ denote the Laplace transform of g. We assume that $\hat{g}$ has the following form

$$\hat{g}(s) = \hat{g}_a(s) + \hat{g}_{ap}(s) + \sum_{\alpha=1}^{\ell} \sum_{m=1}^{m_\alpha} r_{\alpha m}/(s-p_\alpha)^m \tag{3}$$

where

Research sponsored by the National Aeronautics and Space Administration, Grant NGL-05-003-016.

[†] Also with the Belgian National Fund for Scientific Research.

$$\begin{cases} \text{the poles } p_\alpha,\ \alpha = 1, 2, \ldots, \ell \text{ are either real with} \\ \quad \text{real residues} \\ \\ \text{or complex conjugate with complex conjugate residues;} \end{cases} \tag{4}$$

$$\mathrm{Re}\, p_\alpha \geq 0 \text{ for } \alpha = 1, 2, \ldots, \ell; \tag{5}$$

$$g_a(\cdot) \text{ is a real valued function} \in L^1[0,\infty) \tag{6}$$

$$g_{ap}(t) = \sum_{i=0}^{\infty} g_i\, \delta(t-t_i) \tag{7}$$

$$g_i \in \mathbb{R} \quad \text{for } i = 0, 1, 2, \ldots, \tag{8}$$

$$\sum_{i=1}^{\infty} |g_i| < \infty \tag{9}$$

$$0 = t_0 < t_1 < t_2 < \ldots < t_i < \ldots . \tag{10}$$

Note that $g_a(\cdot) + g_{ap}(\cdot)$ belongs to the convolution-algebra $\mathcal{A}$, [3], if and only if its Laplace transform $\hat{g}_a(\cdot) + \hat{g}_{ap}(\cdot)$ belongs to the algebra $\hat{\mathcal{A}}$[†] with pointwise product. Moreover, the function $s \mapsto \hat{g}_{ap}(s)$ mapping $\{s \in \mathbb{C} : \mathrm{Re}\, s \geq 0\}$ into $\mathbb{C}$ is almost periodic in $\mathrm{Re}\, s \geq 0$.

The unperturbed model (nominal in this case) is defined to be F^0, i.e., F^0 is the feedback system model governed by

$$y = g * e \tag{11}$$

$$e = u - y \tag{12}$$

The high-frequency model is a member of the family $(F_H^\varepsilon)_{\varepsilon \geq 0}$ defined by

$$y = T_\varepsilon(g_{ap} * e) \tag{13}$$

$$e = u - y \tag{14}$$

Results

We seek conditions under which the L^2-stability and L^2-instability of the family $(F^\varepsilon)_{\varepsilon \geq 0}$, where $\varepsilon \geq 0$ is sufficiently small, can be deduced from those of F^0 and the family $(F_H^\varepsilon)_{\varepsilon \geq 0}$, where again $\varepsilon \geq 0$ is sufficiently small.

[†] $\hat{\mathcal{A}}$ is the algebra of the Laplace transforms of elements in $\mathcal{A}$.

Lemma

Consider the family of feedback system models $(F_H^{\varepsilon})_{\varepsilon \geq 0}$ defined by (13) and (14). Under these conditions

(A) if $\sup_{\omega \in \mathbb{R}} |\hat{g}_{ap}(j\omega)| < 1$ then F_H^{ε} is L^2-stable for all $\varepsilon \geq 0$,

(B) if $\sup_{\omega \in \mathbb{R}} |g_{ap}(j\omega)| > 1$ then, given any $\mu > 0$, there exists an $\bar{\varepsilon} > 0$ such that $0 < \bar{\varepsilon} < \mu$ and $F_H^{\bar{\varepsilon}}$ is L^2-unstable;

(C) if $\hat{g}_{ap}(j\omega)$ is periodic then F_H^{ε} is L^2-stable for all sufficiently small $\varepsilon \geq 0$ if and only if $\sup_{\omega \in \mathbb{R}} |\hat{g}_{ap}(j\omega)| < 1$.

Theorem:

Consider the two families of feedback system models $(F^{\varepsilon})_{\varepsilon \geq 0}$ defined by (1) and (2) and $(F_H^{\varepsilon})_{\varepsilon \geq 0}$ defined by (13) and (14).

Case 1. Assume $\sup_{\omega \in \mathbb{R}} |\hat{g}_{ap}(j\omega)| < 1$. Under this assumption

(A) if F^0 is L^2-stable then F^{ε} is L^2-stable for all sufficiently small $\varepsilon \geq 0$;

(B) if F^0 is L^2-unstable and if $\begin{cases} 1 + g_0 \neq 0 \\ 1 + \hat{g}(j\omega) \neq 0 \text{ for all } \omega \in \| \end{cases}$

then F^{ε} is L^2-unstable for all sufficiently small $\varepsilon \geq 0$.

Case 2. Assume

either $\sup_{\omega \in \mathbb{R}} |\hat{g}_{ap}(j\omega)| > 1$

or $\sup_{\omega \in \mathbb{R}} |\hat{g}_{ap}(j\omega)| \geq 1$ and $\hat{g}_{ap}(j\omega)$ is periodic

Under this condition, given any $\mu > 0$, there exists an $\bar{\varepsilon} > 0$ such that $0 < \bar{\varepsilon} < \mu$ and $F^{\bar{\varepsilon}}$ is L^2-unstable (whether or not F^0 is L^2-stable).

REFERENCES

[1] C. A. Desoer, "Singular Perturbation and Bounded-Input Bounded-State Stability," Electronics Letters, 6, 16, 1970.

[2] C. A. Desoer and M. J. Shensa, "Networks with very Small and Very Large Parasitics: Natural Frequencies and Stability," IEEE Proc, 58, 12, December 1970, pp. 1933-1938.

[3] C. A. Desoer and M. Y. Wu, "Stability of Linear Time-Invariant Systems," IEEE Trans. on Circuit Theory, CT-15, 3, September 1968, pp. 245-250.

[4] C. A. Desoer and M. Y. Wu, "Stability of Multiple-Loop Feedback Linear Time-Invariant Systems," Jour. Math. Anal. and Appl., 23, June 1968, pp. 121-130.

[5] C. A. Desoer and F. L. Lam, "On the Input-Output Properties of Linear-time Invariant Systems," IEEE Trans. on Circuit Theory, CT-19, 1, January 1972, pp. 60-62.

[6] C. A. Desoer and F. M. Callier, "Convolution Feedback Systems," to appear in SIAM Jour. on Control, 10, 4, November 1972.

[7] C. A. Desoer, "A General Formulation of the Nyquist Criterion," IEEE Trans. on Circuit Theory, CT-12, 2, June 1965, pp. 230-234.

[8] F. M. Callier and C. A. Desoer, "A Graphical Test for Checking the Stability of a Linear Time-Invariant Feedback System," to appear in IEEE Trans. on Automatic Control, AC-17, December 1972.

[9] J. H. Davis, "Encirclement Conditions for Stability and Instability of Feedback Systems with Delays," International Journal of Control, 15, 4, April 1972, pp. 793-799.

[10] J. C. Willems, "The Analysis of Feedback Systems," Research Monograph no. 62, M.I.T. Press, Cambridge, Massachusetts.

[11] H. Bohr, "Almost Periodic Functions," Chelsea, N. Y., 1951.

[12] B. Jessen and H. Thornehave, "Mean Motions and Zeros of Almost Periodic Functions," Acta Mathematics (Upsala), 77, 1945, pp. 137-279. (pp. 143-148, pp. 167-192 esp.).

[13] C. Corduneanu, "Almost Periodic Functions," Interscience, N. Y., 1961.

THE ANALYTIC EVALUATION OF e^{At} USING A COMPUTER

William C. Miller
Maher Sid-Ahmed
Department of Electrical Engineering
University of Windsor
Windsor, Ontario, Canada

ABSTRACT

This paper deals with the problem of generating analytic, rather than numerical, solutions of the matrix exponential, e^{At}, where A is a real n x n matrix. A computer program based on the Cayley-Hamilton theorem has been written.

CALCULATING e^{At}

The program described in this paper was developed as an educational tool for students studying linear systems. In this area of application the matrix A rarely needs to exceed the dimension of 25 x 25. For small scale systems of this nature the Cayley-Hamilton[1] theorem provides a simple method for evaluating e^{At}. It was found, however, that the development of a working program was not as straight forward as originally invisioned.

The accurate determination of the matrix exponential can be difficult if the matrix is badly balanced, that is, if corresponding rows and columns have very different norms. The balancing procedure usually gives a considerable improvement in the accuracy of the computed eigenvalues. Isolated eigenvalues are found exactly no matter how ill conditioned they are. Next, the resultant n x n real matrix is reduced to upper Hessenburg form using stabilized elementary similarity transformations. The eigenvalues of the real upper Hessenberg matrix are then found by the QR method. A sorting procedure is then applied to determine the number of distinct eigenvalues and their multiplicity.

As the matrix exponential has been evaluated using

$$e^{At} = \sum_{k=0}^{n-1} \alpha_k A^k \tag{1}$$

it is necessary to evaluate the α_i terms which are functions of time. Their calculation is complicated by the presence of multiple eigenvalues. A matrix procedure employing a form similar to the transpose of a Vandermonde matrix has been used.

The resultant program prints out the analytic expression for e^{At}. Each element in e^{At} has n terms in it. The program has been written in such a manner that the analytic and the numerical solution for the system of linear equations

$$\frac{d}{dt}[X] = [A][X], \qquad X(0) = X_o \tag{2}$$

is readily available.

EXAMPLE

Compute e^{At} where A is given by

$$A = \begin{bmatrix} 0 & 1 & 3 \\ 6 & 0 & 2 \\ -5 & 2 & 4 \end{bmatrix} \tag{3}$$

The solution to this problem is given by

$$e^{At} = \begin{bmatrix} \left(9e^t + 7te^t - 8e^{2t}\right) & \left(-4e^t - 3te^t + 4e^{2t}\right) & \left(-8e^t - 5te^t + 8e^{2t}\right) \\ \left(22e^t + 28et^t - 22e^{2t}\right) & \left(-10e^t - 12te^t + 11e^{2t}\right) & \left(-22e^t - 20te^t + 22e^{2t}\right) \\ \left(-2e^t - 7te^t + 2e^{2t}\right) & \left(e^t + 3te^t - e^{2t}\right) & \left(3e^t + 5te^t - 2e^{2t}\right) \end{bmatrix} \quad .....(4)$$

A partial portion of the output generated by the program has the form.

```
eAt ( 1,  1) =
     +( 0.90000E 01+j 0.00000E 00)*EXP(( 0.10000E 01+J 0.00000E 00)*T)
  +( 0.70000E 01+J 0.00000E 00)* (T** 1)* EXP(( 0.10000E 01+J 0.00000E00)*T)
      +(-0.80000E 01+J 0.00000E 00)*EXP(( 0.20000E 01+J 0.00000E 00)*T)
```

The computer program was written in FORTRAN IV. In the development of this program some of the subroutines in the eigensystem package, EISPACK[2], from the Argonne National Laboratory were found to be very useful. A program listing is available from the authors.

REFERENCES

1. R. BELLMAN, *"Introduction to Matrix Analysis"*, McGraw-Hill, New York, 1970.
2. ARGONNE CODE CENTER, BLDG.203, ROOM C230, ARGONNE NATIONAL LABORATORY, 9700 S.Cass Avenue, ARGONNE, Illinois, 60439, U.S.A.

SMITH-MCMILLAN FORM FROM STATE-SPACE REPRESENTATION FOR NON-MINIMAL SYSTEMS

O. M. GRASSELLI
Centro di Studio dei Sistemi di Controllo e Calcolo Automatici
Consiglio Nazionale delle Ricerche
Roma, Italy

ABSTRACT

In this paper the Smith-McMillan form of the transfer matrix of a linear, time-invariant system is directly deduced from one of its state-space representations. It is shown that, as in the single-variable case, non-minimality occurs if there are some "cancellations".

Through the theory developed insight is gained into the structure of these systems. As an application, conditions are deduced under which the observability of a controllable system is preserved, or can be achieved, under state-feedback.

1 - INTRODUCTION

A linear, time-invariant system is usually called minimal if its order coincides with the order of a minimal realization of its transfer matrix; i.e. if it is both controllable and observable.

For such a system Rosenbrock has shown how the Smith-McMillan form of its transfer matrix can be directly deduced from one of its state-space representations |1|.

Although the problems connected with realization theory have been widely investigated, also recently (see |1|, |2|, |3| for example), nevertheless an analogous result has not yet been found for the case of non-minimal systems.

In this paper an answer is given to this problem. Through the theory developed insight is gained into the structure of these systems. For example conditions are deduced under which the observability of a controllable system is preserved, or it can be achieved, under state feedback.

2 - NOTATIONS AND TWO BASIC LEMMAS

Consider the linear, time-invariant system S described by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{1}$$

where x, u, y are respectively the state n-vector, the input p-vector, the output q-vector, and A, B, C, D are constant matrices of the proper dimensions. This system will be said to be minimal if it is both controllable and observable. In what follows only constant matrices will be considered, unless otherwise specified.

Define

$$m = \min(p,q) \tag{2}$$

$$P(s) = \begin{bmatrix} sI-A & B \\ -C & D \end{bmatrix} \tag{3}$$

where s is a complex variable and I is the identity matrix; let $n+m_o$ be the

This work was supported by the "Consiglio Nazionale delle Ricerche", Roma.

rank of the polynomial matrix P(s) (which will be called the system matrix of S), and $\alpha_i(s)$ $(i=1,\dots,n+m_o)$ the invariant polynomials of P(s) ($\alpha_i(s)$ divides $\alpha_{i+1}(s)$, $i=1,\dots,n+m_o-1$); if $m_o < m$ define

$$\alpha_{n+i}(s) = 0 \qquad i=m_o+1,\dots,m \qquad (m_o < m) \tag{4}$$

Denote with $\xi_i(s)$ $(i=1,\dots,n)$ the invariant polynomials of sI-A, in such an order that for each $i = 1,\dots,n-1$ $\xi_{i+1}(s)$ divides $\xi_i(s)$; in particular $\xi_1(s)$ is the minimal polynomial of A. If $n < m$ define

$$\xi_i(s) = 1 \qquad i = n+1,\dots,m \qquad (n < m) \tag{5}$$

Moreover let m_o' be the maximum order of non-zero minors of the transfer matrix G(s) of S, which is given by

$$G(s) = C(sI-A)^{-1} B + D \tag{6}$$

Denote with $\varepsilon_i(s)$ and $\psi_i(s)$ $(i=1,\dots,m_o')$ respectively the numerator and denominator polynomials in the Smith-McMillan form of G(s) (ε_i divides ε_{i+1}, ψ_{i+1} divides ψ_i, $i = 1,\dots,m_o'-1$). If $m_o' < m$ define further

$$\varepsilon_i(s) = 0 \qquad i = m_o'+1,\dots,m \qquad (m_o' < m) \tag{7}$$

$$\psi_i(s) = 1 \qquad i = m_o'+1,\dots,m \qquad (m_o' < m) \tag{8}$$

Lastly, denote with n_b the order of a minimal realization of G(s).

Two lemmas will now be given, which will be needed in the development of the theory.

The first one can be proved with the help of the geometric theory of elementary divisors |4|. However, for the sake of brevity, the proofs of both will be omitted.

Lemma 1: For each $j = 1,\dots,m$ $\psi_j(s)$ divides $\xi_j(s)$.

Before stating the second lemma some other symbol definitions are needed.

Given a matrix E, denote with $E_{k_1,\dots,k_r}^{h_1,\dots,h_i}$ the i×r sub-matrix of E remaining after all except rows $h_1,\dots,h_i$ and all except columns $k_1,\dots,k_r$ have been deleted. Furthermore, for each $j = 1,\dots,m$ call $\beta_j(s)$ the greatest common divisor of all the non-zero minors of order n+j of P(s) of the type

$$\det P_{1,\dots,n,n+k_1,\dots,n+k_j}^{1,\dots,n,n+h_1,\dots,n+h_j} \qquad \begin{array}{l} 1\le h_1\le\dots\le h_j\le q \\ 1\le k_1\le\dots\le k_j\le p \end{array} \tag{9}$$

If for an index j no such non-zero minor exists, define $\beta_j(s) = 0$. Lastly define

$$\beta_o(s) = \det P_{1,\dots,n}^{1,\dots,n} \tag{10}$$

and

$$\ell = \min(m_o,n-1) \tag{11}$$

By definition, for each $j = 0,\dots,m_o$ $\beta_j(s)$ is divided by the polynomial $\Pi_{1}^{n+j}{}_{i}\,\alpha_i(s)$. The following lemma gives other properties of the polynomials $\beta_j(s)$.

Lemma 2: a) For each $j = 0,\dots,m_o$ $\beta_j(s)$ is non-zero.

b) For each $j = 0,\dots,\ell$ the polynomial $\Pi_{j+1}^{n}{}_{i}\,\xi_i(s)$ divides $\beta_j(s)$.

Remark 1 - Denoting with $\beta_j'(s)$ the least common multiple of the polynomials $\Pi_{1}^{n+j}{}_{i}\,\alpha_i(s)$ and $\Pi_{j+1}^{n}{}_{i}\,\xi_i(s)$, it follows from Lemma 2 that $\beta_j'(s)$ divides $\beta_j(s)$ $(j=0,\dots,\ell)$. However, in general $\beta_j'(s)$ and $\beta_j(s)$ do not coincide, counterexamples being easily found.◁

Lemma 2 allows the following polynomials to be defined:

$$\gamma_j(s) = \frac{\beta_j(s)}{\Pi_{j+1}^{n}{}_{i}\,\xi_i(s)} \qquad j = 0,\dots,\ell \tag{12}$$

If $m_o \geq n$ define further

$$\gamma_j(s) = \beta_j(s) \qquad j = n,\dots,m_o \qquad (m_o \geq n) \tag{13}$$

As it will be shown in the following section, the polynomials $\gamma_j(s)$, as well as the polynomials $\xi_j(s)$, play a fundamental role in determining the Smith-McMillan form of $G(s)$.

3 - SMITH-MCMILLAN FORM FROM SYSTEM MATRIX

In this section the polynomials $\varepsilon_j(s)$ and $\psi_j(s)$ will be deduced from the polynomials $\gamma_i(s)$ and $\xi_i(s)$, thus giving an answer to the problem that was put forward in the introduction.

In addition a more direct connection between the polynomials $\varepsilon_j(s)$ and the polynomials $\alpha_j(s)$ will be found.

The first result is stated by the following theorem.

Theorem 1: For the system S described by (1):

a) $m_o' = m_o$;

b) if $m_o > 0$, then for each $j = 1,\dots,m_o$ $\gamma_{j-1}(s)$ divides $\gamma_j(s)$;

c) if $m_o > 0$, then, denoting with $\phi_j(s)$ the greatest common divisor of the polynomials $\xi_j(s)$ and $\gamma_j(s)/\gamma_{j-1}(s)$ $(j=1,\dots,m_o)$, the following equations hold:

$$\varepsilon_j(s) = \frac{\gamma_j(s)}{\gamma_{j-1}(s)}\,\frac{1}{\phi_j(s)} \qquad j = 1,\dots,m_o \tag{14}$$

$$\psi_j(s) = \frac{\xi_j(s)}{\phi_j(s)} \qquad j = 1,\dots,m_o \tag{15}$$

Proof. The proof will be carried out deducing the minors of G(s) both from its Smith-McMillan form M(s) and from P(s).

As regards the latter, note that for each j = 1,...,m the minors of order j of G(s) can be written as |1|

$$\det G^{h_1,\dots,h_j}_{k_1,\dots,k_j} = \frac{1}{\det(sI-A)} \det P^{1,\dots,n,n+h_1,\dots,n+h_j}_{1,\dots,n,n+k_1,\dots,n+k_j} \tag{16}$$

Thus, by definition of the polynomials $\beta_j(s)$ and by Lemma 2, m_o represents the maximum order of non-zero minors of G(s). This proves a).

Suppose now that $m_o > 0$. It follows from (16) that, if $j \leq m_o$, then there exists a polynomial in s $g^{h_1,\dots,h_j}_{k_1,\dots,k_j}$ (which is uniquely determined by $h_1,\dots,h_j$, $k_1,\dots,k_j$) such that

$$\det G^{h_1,\dots,h_j}_{k_1,\dots,k_j} = \frac{\beta_j(s)}{\prod_{i=1}^{n} \xi_i(s)} \, g^{h_1,\dots,h_j}_{k_1,\dots,k_j} \tag{17}$$

In addition, for each $j = 1,\dots,m_o$ all the non-zero polynomials $g^{h_1,\dots,h_j}_{k_1,\dots,k_j}$ have 1 as their greatest common divisor.

On the other hand, by definition of M(s), for each $j = 1,\dots,m_o$ the minors of order j of G(s) can be written as

$$\det G^{h_1,\dots,h_j}_{k_1,\dots,k_j} = \prod_{i=1}^{j} \frac{\varepsilon_i(s)}{\psi_i(s)} \, g'^{h_1,\dots,h_j}_{k_1,\dots,k_j} \tag{18}$$

where $g'^{h_1,\dots,h_j}_{k_1,\dots,k_j}$ is a polynomial which is uniquely determined by $h_1,\dots,h_j$, $k_1,\dots,k_j$. In addition for each $j = 1,\dots,m_o$ all the non-zero polynomials $g'^{h_1,\dots,h_j}_{k_1,\dots,k_j}$ have 1 as their greatest common divisor.

From (17), (18) and the above discussion it is easily deduced that

$$\prod_{i=1}^{j} \frac{\varepsilon_i(s)}{\psi_i(s)} = \frac{\beta_j(s)}{\prod_{i=1}^{n} \xi_i(s)} \qquad j = 1,\dots,m_o \tag{19}$$

From (10) and (19) it follows that

$$\frac{\varepsilon_j(s)}{\psi_j(s)} = \frac{\beta_j(s)}{\beta_{j-1}(s)} \qquad j = 1,\dots,m_o \tag{20}$$

From (5), (11), (12), (13), (20)

$$\frac{\varepsilon_j(s)}{\psi_j(s)} = \frac{\gamma_j(s)}{\gamma_{j-1}(s)} \frac{1}{\xi_j(s)} \qquad j = 1,\dots,m_o \tag{21}$$

Thus, b) follows from Lemma 1 and (21).

Furthermore, remembering that $\varepsilon_j(s)$ and $\psi_j(s)$ are co-prime, (14) and (15) are deduced from the definition of $\phi_j(s)$ and from (21). This completes the proof. ◁

Corollary 1: If $m_o > 0$, then

$$\frac{\varepsilon_j(s)}{\psi_j(s)} = \frac{\beta_j(s)}{\beta_{j-1}(s)} \qquad j = 1,\dots,m_o \tag{22}$$

Corollary 2: S is minimal if and only if the following conditions are satisfied:

i) $m_o > 0$;

ii) $\phi_j(s) = 1, \quad j = 1,\dots,m_o$;

iii) if $m_o < n$, then

$$\xi_j(s) = 1 \qquad j = m_o+1,\dots,n \tag{23}$$

Proof. By definition S is minimal if and only if n_b=n. Remember that n_b coincides with the degree of the polynomial $\Pi_1^m{}_i \ \psi_i(s)$ |1|, and that n coincides with the degree of the polynomial $\Pi_1^n{}_i \ \xi_i(s)$. Thus the corollary follows from (5), (8) and (15). ◁

Corollary 3: S is minimal if and only if the following conditions are satisfied:

i) $m_o > 0$;

ii) $\gamma_j(s) = \Pi_1^j{}_i \ \alpha_{n+i}(s), \quad j = 1,\dots,m_o$;

iii) for each $j = 1,\dots,m_o$ $\alpha_{n+j}(s)$ and $\xi_j(s)$ are co-prime;

iv) if $m_o < n$, then

$$\xi_j(s) = 1 \qquad j = m_o+1,\dots,n \tag{24}$$

Proof. The corollary follows directly from Theorem 1 and Corollary 2, taking into account that minimality implies |1|

$$\varepsilon_j(s) = \alpha_{n+j}(s) \qquad j = 1,\dots,m \quad \triangleleft \tag{25}$$

Remark 2 - Theorem 1 and Corollary 2 show the meaning of the polynomials $\gamma_j(s)/\gamma_{j-1}(s)$ and $\phi_j(s)$. The zeros of the polynomials $\varepsilon_j(s)$ (which can be called the "zeros of the minimal part of S") are a sub-set of the zeros of the polynomials $\gamma_j(s)/\gamma_{j-1}(s)$, as well as the zeros of the $\psi_j(s)$ (i.e. the "poles" of the minimal part of S) are a sub-set of the zeros of the $\xi_j(s)$ (i.e. the poles of S). Non-minimality of S occurs if for some j $\gamma_j(s)/\gamma_{j-1}(s)$ and $\xi_j(s)$ have common factors (the zeros of the polynomials $\phi_j(s)$ represent the poles of S which are "canceled", i.e. the poles of the non-minimal part of S), as well as if the number of non-constant polynomials $\xi_j(s)$ is greater than the number of non-zero polynomials $\beta_j(s)$ (also this case can be seen as a cancellation). This generalizes the well-known minimality condition valid for single-variable systems.◁

Remark 3 - If S is minimal, then

$$\beta_j(s) = \left[\prod_{1}^{j}{}_i\ \alpha_{n+i}(s)\right]\left[\prod_{j+1}^{n}{}_i\ \xi_i(s)\right] \qquad j = 1,\dots,\ell \tag{26}$$

and, if $m_o \geq n$,

$$\beta_j(s) = \prod_{1}^{j}{}_i\ \alpha_{n+i}(s) \qquad j = n,\dots,m_o \qquad (m_o \geq n) \tag{27}$$

Moreover it is easily seen that the controllability or the observability of S imply

$$\alpha_j(s) = 1 \qquad j = 1,\dots,n \tag{28}$$

Thus, it follows from Corollary 3 and from the definition of $\beta'_j(s)$ (see Remark 1) that, if S is minimal, then

$$\beta_j(s) = \beta'_j(s) \qquad j = 0,\dots,\ell \quad \triangleleft \tag{29}$$

The following theorem points out more directly than Theorem 1 the connection between the polynomials $\varepsilon_i(s)$ and the polynomials $\alpha_i(s)$.

Theorem 2: If $m_o > 0$, then for each $j = 1,\dots,m_o$ the polynomial $\prod_{1}^{j}{}_i\ \varepsilon_i(s)$ divides the polynomial $\prod_{1}^{n+j}{}_i\ \alpha_i(s)$.

Proof. If $n_b = 0$, then $G(s) = D$ and the theorem is obvious.

Then assume that $n_b > 0$, and consider a canonical representation of S |5|: let T be a non-singular square matrix such that

$$T^{-1}AT = \begin{bmatrix} A_{aa} & A_{ab} & A_{ac} & A_{ad} \\ 0 & A_{bb} & 0 & A_{bd} \\ 0 & 0 & A_{cc} & A_{cd} \\ 0 & 0 & 0 & A_{dd} \end{bmatrix} \tag{30}$$

$$T^{-1}B = \begin{bmatrix} B_a \\ B_b \\ 0 \\ 0 \end{bmatrix} \tag{31}$$

$$CT = \begin{bmatrix} 0 & C_b & 0 & C_d \end{bmatrix} \tag{32}$$

where A_{bb} is the $n_b \times n_b$ dynamic matrix of the controllable and observable sub-system of S, and the meaning of the undefined symbols is obvious.

Moreover call n^c the order of the controllable sub-system of S; if $n^c > n_b$ (i.e. if A_{aa} does not vanish) define

$$P^c(s) = \begin{bmatrix} sI - A_{aa} & \begin{bmatrix} A_{ab} & B_a \end{bmatrix} \\ 0 & P_b(s) \end{bmatrix} \tag{33}$$

where

$$P_b(s) \triangleq \begin{bmatrix} sI - A_{bb} & B_b \\ -C_b & D \end{bmatrix} \tag{34}$$

Point a) of Theorem 1 implies that $P^c(s)$ has rank $n^c + m_o$. Denote with $\alpha_i^c(s)$ $(i=1,\dots,n^c+m_o)$ the invariant polynomials of $P^c(s)$ ($\alpha_i^c(s)$ divides $\alpha_{i+1}^c(s)$, $i = 1,\dots,n^c+m_o-1$).

The proof will be carried out into two steps. Firstly it will be shown that for each $j = 1,\dots,m_o$ $\prod_1^j \varepsilon_i(s)$ divides $\prod_1^{n^c+j} \alpha_i^c(s)$; then, that for each $j = 1,\dots,m_o$ $\prod_1^{n^c+j} \alpha_i^c(s)$ divides $\prod_1^{n+j} \alpha_i(s)$.

As regards the former statement, there certainly exist square polynomial matrices L(s) and R(s) with constant non-zero determinants (respectively of dimensions n_b+q and n_b+p) and a polynomial matrix $Q_1(s)$ such that

$$\begin{bmatrix} I & 0 \\ 0 & L(s) \end{bmatrix} P^c(s) \begin{bmatrix} I & 0 \\ 0 & R(s) \end{bmatrix} = \begin{bmatrix} sI - A_{aa} & Q_1(s) \\ 0 & Q(s) \end{bmatrix} \tag{35}$$

where Q(s) is the Smith form of $P_b(s)$, which is given by |1|

$$Q(s) = \begin{bmatrix} I & 0 & 0 \\ 0 & \mathrm{diag}(\varepsilon_1(s),\dots,\varepsilon_m(s)) & 0 \end{bmatrix} \quad (m = q) \tag{36}$$

or by

$$Q(s) = \begin{bmatrix} I & 0 \\ 0 & \mathrm{diag}(\varepsilon_1(s),\dots,\varepsilon_m(s)) \\ 0 & 0 \end{bmatrix} \quad (m = p) \tag{37}$$

Thus the first statement follows from (35), (36), (37). Moreover the second statement can be deduced in a similar way from (30), (31), (32). This completes the proof. ◁

Remark 4 - The connection between the polynomials $\prod_1^j \varepsilon_i(s)$ and the polynomials $\prod_1^{n+j} \alpha_i(s)$ which is shown by Theorem 2 is similar to the connection between the $\psi_j(s)$ and the $\xi_j(s)$ which is shown by Lemma 1; the zeros of the $\varepsilon_i(s)$ (i.e. the "zeros of the minimal part of S") are a sub-set of the zeros of the $\alpha_i(s)$. ◁

Remark 5 - The theory developed gives a direct connection between the Smith-McMillan form of the transfer matrix of S and its state-space representation (1). However, the results here obtained can be easily generalized to the case in which the description of S has not been brought into the state-space form (1), but is constituted by a set of linear equations of the type

$$\begin{aligned} T(s)\, z &= U(s)\, u \\ y &= V(s)\, z + W(s)\, u \end{aligned} \tag{38}$$

(Laplace transforms have been taken), where T(s), U(s), V(s), W(s) are polynomial matrices, T(s) being also square and with a non-zero determinant of degree n. ◁

4 - AN APPLICATION TO STATE FEEDBACK

As an application of the theory developed in last section, consider for the system S the control law given by

$$u = F x + u_o \tag{39}$$

where the matrix F represents a state feedback. The closed-loop system described by (1) and (39), having u_o as input, will be denoted with S^F. Similarly, all the symbols previously considered referring to S^F will be denoted with a superscript F.

It is well-known that S^F is controllable if and only if S is controllable |6|, while observability may or may not be preserved.

A condition under which, given the controllability of S, S^F is observable for any F, will now be deduced from Theorem 1 and its corollaries and from the following lemma (which is easily shown by an approach similar to the proof of Theorem 2).

Lemma 3: If S is controllable, and also the polynomials $\prod_1^{n+m_o} \alpha_i(s)$ and $\prod_1^{n} \xi_i(s)$ are co-prime and rank $\begin{bmatrix} B \\ D \end{bmatrix} = m_o$, then S is observable.

Theorem 3: If S is controllable and D = 0, then S^F is observable for any F if and only if

$$\text{rank}\, B = m_o \tag{40}$$

$$\alpha_{n+m_o}(s) = 1 \tag{41}$$

Proof. The system matrix of S^F is given by

$$P^F(s) = \begin{bmatrix} sI - A - BF & B \\ - C - DF & D \end{bmatrix} \tag{42}$$

From (42) it is easily seen that

$$\alpha_j^F(s) = \alpha_j(s) \qquad j = 1,\dots,n+m \qquad \forall F \tag{43}$$

disregarding any hypothesis on S; this implies $m_o^F = m_o$.

As regards the necessity, note that

$$m_o \leq \text{rank}\, B \leq n \tag{44}$$

since D = 0. Hence if $m_o = n$ (40) follows. If $m_o < n$, from the minimality of S^F and from Corollary 2 it follows that

$$\xi_j^F(s) = 1 \qquad j = m_o+1,\dots,n \qquad \forall F \tag{45}$$

what implies, by the controllability of S |1|,

$$\text{rank}\, B \leq m_o \tag{46}$$

This proves (40) in the case $m_o < n$.

It follows from the controllability of S and from (40) that there certainly exists a matrix F such that the polynomial $\xi_{m_o}^F(s)$ has any assigned zero |1|. By Corollary 3 the minimality of S^F and (43) imply (41).

The sufficiency is directly deduced from Lemma 3. This completes the proof. ◁

Remark 6 - The controllability of S and (41) are not sufficient for the observability of S^F (counterexamples being easily found).

In the general case, given the controllability of S, S^F may or may not be observable. However Lemma 3 implies that, if rank$\begin{bmatrix} B \\ D \end{bmatrix} = m_o$, then S^F is observable if F is chosen such that the polynomials det(sI-A-BF) and $\prod_{1}^{n+m_o} \alpha_i(s)$ are co-prime. ◁

Remark 7 - From Theorem 2 and from (43) severe restrictions follow on the possibility of obtaining assigned $\varepsilon_i^F(s)$.

5 - CONCLUSIONS

Through the theory developed in this paper insight is gained into the structure of linear, time-invariant systems. Furthermore this theory can be seen as a first step towards deducing the Smith-McMillan form of the series connection of two systems from the state-space representations of both. Such a result should be helpful in studying properties of composite systems, for example whenever attention is paid to minimality.

REFERENCES

|1| Rosenbrock, H.H., "State-Space and Multivariable Theory", Nelson, London, 1970.

|2| Heymann, M., "Transfer Equivalence of Linear Dynamical Systems", SIAM J. Control, Vol. 8, n. 1, February 1970.

|3| Heymann, M., "Lectures on Structure and Realization Problems in the Theory of Dynamical Systems", Lectures delivered at the International Center for Mechanical Sciences, Udine, Italy, July 1972.

|4| Gantmacher, F.R., "The Theory of Matrices", Chelsea, New York, Vol. 1, 1959.

|5| Kalman, R.E., "Mathematical Description of Linear Dynamical Systems", SIAM J. on Control, Vol. 1, p. 152-192, 1963.

|6| Wonham, W.M., "On Pole Assignment in Multi-Input Controllable Linear Systems", IEEE Transactions on Automatic Control, Vol. AC-12, n. 6, December, 1967.

ON THE DETERMINATION OF THE CAUSAL PART OF A NONCAUSAL SYSTEM*

R.M. DESANTIS
Département de Génie Electrique
Ecole Polytechnique
Montréal, P.Q., Canada

ABSTRACT

This article is about the causality structure of systems which can be modelled as operators on a Hilbert space equipped with a resolution of the identity. Most of the linear results presently available on the subject are extended into a nonlinear systems setting.

INTRODUCTION

To help introduce subject and terminology, let $L_2(-\infty, +\infty)$ denote the Hilbert space of square integrable real valued functions defined on the real line, and let a system T be such that its input and output can be put in correspondence with functions in $L_2(-\infty,+\infty)$, and if $x \in L_2(-\infty, +\infty)$, and $y = Tx$, then y can be computed by the equation

$$y(t) = \int_{-\infty}^{+\infty} K(t,\tau)x(\tau)d\tau, \quad t \in (-\infty,+\infty), \tag{1}$$

where $K(t,\tau)$ has the property that

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} K^2(t,\tau)dtd\tau < \infty. \tag{2}$$

Unless one makes the hypothesis that $K(t,\tau) = 0$ for $\tau > t$ and almost all $t \in (-\infty, +\infty)$, the value of y at a given time t is a function of past and future values of x. Accordingly, T is said to be physically nonrealizable or noncausal. On the other hand, one observes that T can be viewed as the sum of two systems T_1 and T_2 defined respectively as follows: if x,w and $z \in L_2(-\infty, +\infty)$, and $w = T_1x$, $z = T_2x$, then

$$w(t) = \int_{-\infty}^{+\infty} K(t,\tau)u(t-\tau)x(\tau)dt, \tag{3}$$

and

$$z(t) = \int_{-\infty}^{+\infty} K(t,\tau)u(\tau-t)x(\tau)dt, \tag{4}$$

where the function u(.) is such that $u(s) = 1$ if $s>0$, and $u(s) = 0$ for $s < 0$.

---

* This study has been supported in part by the U.S. National Science Foundation, Grant GK 1925, and by the Canadian National Research Council, Grant A-8244.

In this case the output of the system T_1 at a given time t is a function only of the past values of the input, and therefore T_1 is physically realizable or causal. In regard to T_2, though the present value of its output is a function of future values of the input, a simple reversal in the order in which the time flows would allow this system also to be viewed as a causal one. A system with the property enjoyed by T_2 is called anticausal, and a system which, like T, can be thought as given by the sum of a causal plus an anticausal system is said to admit a causality decomposition. The components of this decomposition are usually referred to as the causal and anticausal parts of the original system.

Though most of the linear systems of engineering interest, do not enjoy the property that their input output pair satisfies an equation type (1), it may still be important to consider the question about the meaning to be given to a causal part, or to a causality decomposition for these more general systems. Does in this case a causality decomposition always exist? If it does exist, is it unique and in what sense? What can one say about nonlinear systems?

The main objective of the paper is to discuss the above questions. This is done by simultaneously treating linear and nonlinear systems. Moreover, in order to make the development applicable not only to systems in $L_2(-\infty,+\infty)$, but also to systems defined on other Hilbert spaces, possibly more complex or less familiar than $L_2(-\infty,+\infty)$, we will use the mathematical framework of Hilbert resolution spaces. This framework turns out to be very convenient to mathematically formalize the concepts already introduced, as well as a number of other additional concepts which will be required. This formalization, in turn, allows one to state necessary and sufficient conditions for a causality decomposition to exist and to be unique. Using these conditions, it becomes then possible to show that for systems in Hilbert spaces, a causality decomposition may fail to exist. This can happen, in particular, in the case of linear systems defined in $L_2(-\infty,+\infty)$, $L_2[0,+\infty)$, or $L_2[a,b]$.

The questions considered in the present development are not new, and have already received considerable attention by the technical literature. In particular, in the context of linear and time invariant systems, they have led, for instance, to the spectral factorization techniques adopted by Wiener [8]. The study of the more general nonlinear and time variant case, however, is more recent. In this regard, Saeks [7]has considered the causality structure of linear systems in a setting which is very similar to ours, and Porter [6]has shown how that study can be extended into a Banach space context. The attention of the reader is also addressed to [2], where the development unfolds in a group valued function space framework and has many points in commun with the present one.

HILBERT RESOLUTION SPACES

Suppose that H is a Hilbert space and ν a linearly ordered set with t_o and t_∞ respectively minimum and maximum elements of ν. A family IR = $\{P^t\}, t \in \nu$, of orthogonal projectors on H is a Resolution of the identity when it enjoys the following two properties:

Ri) $P^{t_o}H = 0$, $P^{t_\infty}H = H$, and $P^k H \geq P^\ell H$ whenever $k > \ell$; Rii) if $\{P^i\}$is a sequence of orthogonal projectors in IR and there exists an orthogonal projector P such that $\{P^i x\} \to Px$, for every $x \in H$, then $P \in$ IR.

A Hilbert space, H, equipped with a Resolution of the identity, I IR = $\{P^t\}$, is called a Hilbert resolution space (in short: HRS) and is denoted by the symbol $[H, P^t]$.

Next, the notion of integral on HRS is introduced. Suppose that T(s), $s \in \nu$, is a family of operators on a HRS indexed by $s \in \nu$, and consider the following operations:

i) Choose a partition Ω of ν, $\Omega = \{\xi_o, \xi_1, \ldots, \xi_N\}$, where $\xi_o = t_o$, $\xi_n = t_\infty$ and $\{\xi_j\}$is a finite set of elements in ν such that $\xi_j < \xi_{j+1}$, $j = 1, 2, \ldots, N-1$;

ii) Consider the partial sum

$$I^\Omega = \sum_{k=1}^{N} \Delta P(\xi_k) T(s_k) \tag{5}$$

where $\Delta P(\xi_k) = P^{\xi_k} - P^{\xi_{k-1}}$ and s_k is any element of ν such that $\xi_{k-1} \leq s_k \leq \xi_k$;

iii) On the set of all partitions Ω of ν, define a partial order as follows: $\Omega_1 \geq \Omega_2$ if every element of Ω_2 is contained in Ω_1;

iv) Suppose that there exists an operator T such that in correspondence to any $\varepsilon > 0$ and $x \in H$, there is a partition Ω_ε of ν with the following property: $|Tx - I^\Omega x| < \varepsilon$ if $\Omega \geq \Omega_\varepsilon$.

The operator T obtained through operations i - iv) is called the integral of the family T(s) with respect to IR and is denoted by $T = \int dPT(s)$.

It is useful to consider slightly different variations of the above concept of integral. To this purpose, the notations ${\int\!\!\!\!\!\!m}\,dPT(s)$ and ${\int\!\!\!\!\!\!M}\,dPT(s)$ will be used to indicate the integrals which are obtained by choosing s_k in operation ii) respectively as follows: $s_k = \xi_{k-1}$ or $s_k = \xi_k$. Similarly, the operator $\int dPT(s)dP$ will denote the integral which is obtained by replacing Equation (5) in operation ii) by the following:

$$I^{\Omega} = \sum_{k=1}^{N} \Delta P(\xi_k) T(s_k) \Delta P(\xi_k).$$

BASIC DEFINITIONS

The following causality related concepts are at the basis of the present development. The reader familiar with previous treatments of the subject will note that the definitions of causality, strict causality, and memorylessness, are formally identical to those proposed by Porter [6], and Saeks [7]. In regard to strong causality, crosscausality, and strong crosscausality, these concepts are a natural extension of those formulated in [2].

Definition 1. T is causal (anticausal) if $P^t x = P^t y$ implies $P^t Tx = P^t Ty$ ($P_t x = P_t y$ implies $P_t Tx = P_t T_y$).

Definition 2. T is memoryless if it is simultaneously causal and anticausal.

Definition 3. T is strongly causal (strongly anticausal) if T is causal (anticausal) and

$$\int dPTdP = 0.$$

Definition 4. T is strictly causal (strictly anticausal) if

$$T = {\int\!\!\!\!\!\!m}\,dPTP^s \left(= {\int\!\!\!\!\!\!M}\,dPTP_s \right) \text{ and } \int dPTdP = 0.$$

Definition 5. T is crosscausal if $\int dPTdP = 0$; $P^t x = 0$ implies $P^t Tx = 0$; and $P_t x = 0$ implies $P_t Tx = 0$.

Definition 6. T is strongly crosscausal if

$${\int\!\!\!\!\!\!M}\,dPTP^s = {\int\!\!\!\!\!\!m}\,dPTP_s = \int dPTdP = 0.$$

In analogy with the procedure adopted in [2], it will be helpful to use the following causality alphabetic code $\mathbf{A} = \{A, C, M, X, \underline{A}, \underline{C}, \underline{X}, \overline{A}, \overline{C}\}$, where

A = anticausal
C = causal
M = memoryless
X = crosscausal
$\underline{A}$ = strongly anticausal.
$\underline{C}$ = strongly causal
$\underline{X}$ = strongly crosscausal
$\overline{A}$ = strictly anticausal
$\overline{C}$ = strictly causal

CAUSALITY PROPERTIES

This section presents some results about causality concepts, invariance of various families of subspaces and integral representations $^{(i)}$.

Proposition 1. The following statements are equivalent:

T is causal; $P^t = P^t P T^t$; $T = \int\!\!\!\!\!\mathrm{M}\, dPTP^s$.

Proposition 2. The following statements are equivalent:

T is memoryless; $P^t T = P^t T P^t$ and $P_t T = P_t T P_t$; $T = \int dPTdP$.

Proposition 3. The following statements are equivalent:

T is crosscausal; $TP_t = P_t T P_t$, $TP^t = P^t T P^t$ and $\int dPTdP = 0$;

$\int\!\!\!\!\!\mathrm{m}\, dPTP^s = \int\!\!\!\!\!\mathrm{M}\, dPTP_s = \int dPTdP = 0$.

Proposition 4. If T is simultaneously α_1 and α_2, with $\alpha_1, \alpha_2 \in \{\underline{A}, \underline{C}, M, \underline{X}\}$, then, either $\alpha_1 = \alpha_2$ or T is the null operator.

Proposition 5. If T is strongly causal (strictly causal) and T" is bounded and causal, then T'T" and T"T' are strongly causal (strictly causal).

Theorem 1. The classes of bounded causal, bounded strongly causal and bounded crosscausal systems are Banach algebras.

The following two propositions indicate that in the case of linear systems the concept of crosscausality disappears and the concept of strong causality is identical to that of strict causality. Thus, in the context of linear systems, the causality framework proposed in this paper is in perfect agreement with that already adopted in [6]and [7] $^{(ii)}$.

Proposition 6. If a system is linear and crosscausal, then it is null.

Proposition 7. A linear system is strongly causal if and only if it is strictly causal.

Additional connections about linearity and causality structure are illustrated by the following results.

Proposition 8. A necessary and sufficient condition for a causal linear system T to be strictly causal is that $\int dPtdP = 0$.

(i) The proofs of all the results presented in this paper can be found in [1]and [3].

(ii) Most of the following considerations about linear systems are also applicable to "weakly additive" systems, a system T being weakly additive if $T = TP^t + TP_t$ for all $t \in \nu$.

Proposition 9.(iii) If T is a linear system then the following statements are equivalent:

$$T \text{ is causal}; \quad P^t x = 0 \text{ implies } P^t Tx = 0; \; TP_t = P_t TP_t$$

$$P^t T = P^t TP^t; \quad T = \mathrlap{\int}M dPTP^s.$$

Proposition 10. If T is a linear system, then the following statements are equivalent:

T is memoryless; $P^t x = 0$ implies $P^t Tx = 0$, and $P_t x = 0$ implies $P_t Tx = 0$; $TP^t = P^t TP^t$ and $TP_t = P_t TP_t$; $P^t T = P^t TP^t$ and $P_t T = P_t TP_t$; $T = \int dPtdP$.

Theorem 2. The classes of bounded causal, and bounded strictly causal linear systems are Banach Algebras.

CANONICAL CAUSALITY DECOMPOSITION

This section contains the main results of paper and considers the question about whether a system on a Hilbert resolution space might be represented by the sum of systems of type $\alpha \in \{\underline{A}, \underline{C}, M, \underline{X}\}$. In line with Saeks' terminology, this representation will be called a canonical causality decomposition.

Theorem 1. A system T has a canonical causality decomposition if and only if the components of this decomposition can be computed as follows:

$$T_M = \int dPTdP; \quad T_{\underline{A}} = \mathrlap{\int}m dP[T - T_M]P_s;$$

$$T_{\underline{C}} = \mathrlap{\int}M dP[T - T_M]P^s; \; T_{\underline{X}} = T - T_{\underline{A}} - T_{\underline{C}} - T_M.$$

Corollary 1. If T has a canonical causality decomposition, then this decomposition is unique.

The importance of Theorem 1 lies in the fact that it links the study of existence of a causality canonical decomposition to the study of the convergence properties of a special class of integrals in HRS. This latter study has recently received attention from a number of Russian mathematicians. Of particular interest here are the following two results from Gohberg and Krein [5].

Lemma 1. ([5], p.26). If T is a linear and bounded operator on a HRS, and $\int P^s TdP$ is well defined, then $\int dPTP^s$ is well defined and

$$\int dPTP^s = T - \int P^s TdP.$$

Moreover, $\int dPTdP = 0$ and

$$\mathrlap{\int}M dPTP^s = \mathrlap{\int}m dPTP^s = \int dPTP^s.$$

(iii) Propositions 9 and 10 were first stated in [7].

Lemma 2. ([5], p. 104). For any orthogonal system $\Phi_j (j = 0, \pm 1, \pm 2, \ldots)$ in a Hilbert space H, there exists a resolution of the identity IR such that for every self-adjoint operator

$$T = \sum_{j=-\infty}^{+\infty} \lambda_j (\cdot, \Phi_j)\Phi_j$$

satisfying the conditions

$$\lambda_j \geq 0 (j = 1,2,\ldots,), \lambda_j \leq 0 (j = 0,-1,\ldots,)$$

$$\sum_{j=-\infty}^{+\infty} \lambda_j /_{2j-1} = \infty,$$

the integral $Y = \int P^s T dP$ diverges in the strong sense. Conversely for any continuous resolution of the identity $IR = \{P^t\}$ there exists an orthonormal system $\Phi_j (j = 0, \pm 1, \ldots)$ such that the above conclusions hold for the specified class of self-adjoint operators.

Using lemma 1 and 2 one can easily obtain the following result.

Proposition 1. There exist systems on a HRS which do not have a canonical decomposition.

In the case of linear systems the above results can be specialized as follows.

Theorem 2. A linear system T has a canonical causality decomposition if and only if the components of this decomposition can be computed as follows:

$$T_{\underline{C}} = \rlap{m}\int dPTP^s \ , \ T_{\underline{A}} = \rlap{M}\int dPTP_s, \text{ and } T_M = \int dPTPdP.$$

Proposition 2. There exist bounded linear systems on a HRS which do not have a canonical causality decomposition.

Proposition 2 shows that even in the context of linear systems, the canonical decomposition question is not as simple as it was thought in an early study [7]. The following two propositions shed some further light on this question.

Proposition 3. Suppose that T is a bounded linear operator and that $T_M = \int dPTdP$ is well defined. Then, given any $\epsilon > 0$, there exists an operator $\tilde{T}$ such that $|T - \tilde{T}| < \epsilon$ and $\tilde{T}$ has a canonical causality decomposition.

Proposition 4. Suppose that a HRS has a continuous resolution of the identity. Then given any operator T and any real $\epsilon > 0$, there exists an operator $\tilde{T}$ such that $|T - \tilde{T}| < \epsilon$ and T does not have a canonical causality decomposition.

CAUSALITY DECOMPOSITION AND HILBERT SCHMIDT OPERATORS

In this section some causality properties related to Hilbert Schmidt operators on a HRS are presented. This is done in view of the importance of this type of operators in control and communication theory, and the fact that the causality properties of these operators enjoy a completeness which is not present in the general case.

To begin with, suppose that T is a bounded linear operator on $[H,P^t]$ and let $\{x_\beta\}$, $\beta\in E$, be a complete orthonormal set in H.

Definition 1. A bounded linear operator T is a Hilbert Schmidt (in short HS) operator if the quantity

$$\|T\| = \{\sum_{\beta\in E} |Tx_\beta|^2\}^{\frac{1}{2}}$$

is finite. The number $\|T\|$ is called the Hilbert Schmidt norm of T.

The following two lemmas state the two special properties of HS operators which determine their particular structure with respect to causality.

Lemma 1. The Hilbert Schmidt norm is independent of the orthonormal basis used in its definition. The set of all HS operators is a Banach space under the Hilbert Schmidt norm.

Lemma 2. The Banach space of HS operators is a Hilbert space. The inner product between two elements T, $S\in HS$ can be defined by

$$< S,T > = \sum_{\beta\in E} < Sx_\beta, Tx_\beta >.$$

The causality structure of HS operators can now be clarified by the following propositions.

Proposition 1. The class of causal (anticausal, memoryless) HS operators is a Hilbert space.

Proposition 2. The space of HS operators is given by the direct sum of the three orthogonal subspaces of strictly causal, strictly anticausal and memoryless operators.

Proposition 3. Suppose that T is a HS operator. Then there exists a unique HS operator $T_\alpha, \alpha\in\{A,C,M,\bar{A},\bar{C},\}$ such that

$$\|T - T_\alpha\| = \min_{\tilde{T}_\alpha} \|T - \tilde{T}_\alpha\|$$

where $\tilde{T}_\alpha$ is any α HS operator and T_α is the α component of T.

The combination of the statements of Propositions 1 and 2 gives the following theorem.

Theorem 1. Every Hilbert Schmidt operator has a causality canonical decomposition. The components of this decomposition are mutually orthogonal.

CONCLUSIONS

The causality development of the present article, can be summarized from two points of view: extension to a Hilbert space context of the study reported in [2]; and generalization to nonlinear systems of the study reported in [6] and [7].

From the first point of view, the paper shows how techniques and results in group valued function spaces can be respectively applied and formulated in Hilbert spaces. This is done via a conceptual correspondence between the role played in [2]by the "sum" type transformators and the equivalent role that integral transformators can play in Hilbert space. This correspondence allows to apply to the present causality study recently developed Volterra operators theory techniques. In particular, these techniques can be used to give necessary and sufficient conditions for the existence and uniqueness of a canonical causality decomposition in Hilbert space.

In regard to the second point of view, the paper shows that if a conceptual framework slightly more involved than that proposed in [6] and [7] is used, then those developments can in a natural and meaningful way be generalized into a nonlinear systems setting. The framework proposed here includes, together with causality and strict causality, the novel concepts of crosscausality and strong causality. When one concentrates its attention to linear operators, then crosscausality vanishes and strong causality is equivalent to strict causality. Thus, in this particular case, the present approach is in perfect agreement with that of [6] and [7]. It is interesting to observe, however, that this agreement cannot be extended to one of the main results in [7], namely Theorem 2.2.3. This is due to the fact that the properties of integral transformators are not as extensive as those which would be needed for the statement of that result to hold. This statement, on the other hand, can be proved valid if its application is confined to special classes of operators. Thus in the case of Hilbert Schmidt operators, for example, a canonical causality decomposition always exists and has mutually orthogonal components.

ACKNOWLEDGEMENT

Helpful suggestions and comments from the author's PhD thesis committee chairman Professor William A. Porter, and from Professor Richard Saeks, are gratefully acknowledged.

REFERENCES

1. DeSantis, R.M., Causality Structure of Engineering Systems. Ph. D. Thesis, The University of Michigan, December 1971.
2. DeSantis, R.M., and Porter, W.A., On time Related Properties of Nonlinear Systems, SIAM J. on Applied Mathematics, Vol. 24, No.2 (to appear).
3. DeSantis, R.M., Causality for Nonlinear Systems in Hilbert Space, Submitted to Math Systems Theory Journal, April 1972.
4. Duttweiler, D.L., Reproducing Kernel Hilbert Space Techniques for Dectection and Estimation Problems. Technical Reprot No. 7050-18, Information Systems Laboratory, Stanford, 1970.
5. Gohberg, I.Z., and Krein, M.G., Theory of Volterra Operators in Hilbert Space and Applications. American Mathematical Society, Vol 24, Providence, Rhode Island, 1970.
6. Porter, W.A., and Zahm, C.L., Basic Concepts in Systems Theory. Technical Report No. 44, Systems Engineering Laboratory, The University of Michigan, 1969.
7. Saeks, R., Causality in Hilbert Space, Siam Review, Vol. 12, No. 3, July 1970.
8. Wiener, N., Extrapolation Interpolation and Smoothing of Sationary Time Series, John Wiley and Sons, Inc., New York, 1949.

ON THE OPTIMAL FEEDBACK CONTROLLERS FOR SYSTEMS OF NEUTRAL TYPE

S. G. Chow H. N. KOIVO
Department of Electrical Engineering
University of Toronto
Toronto, Canada

ABSTRACT

The powerful Fredholm resolvent theory is utilized to obtain the optimal feedback solution for a linear equation of neutral type with a quadratic cost, in a compact form, which also provides more insight to the structure than the earlier solution appearing in the literature.

INTRODUCTION

The optimization of time delay systems has extensively been explored in the recent control literature. A good survey of the area can be found in [2, 10, 12].

The mathematical theory of neutral systems in which the present rate of change of a state depends on the past rate of change as well as the past and present values of the states has been developed in [4, 7, 8]. Physical motivation for the study of such systems arises, for example, in the study of collision problem in electrodynamics [6], in the theory of lossless transmission lines [5], and in the study of vibrating masses attached to elastic bars [16].

Optimal control theory of such systems has been developed to some extent. A maximum principle for these systems is derived in [9]. The method of Neustadt [14] can be utilized to derive necessary conditions for a large class of functional equations of neutral type [3].

The optimal synthesis of linear differential equations of neutral type with a quadratic cost is considered in [1] by using dynamic programming approach. However, the result is in a somewhat awkward form involving many coupled partial differential equations on each unit interval (assuming the delay to be a unit), thus rendering the actual numerical solution of such equations quite cumbersome. Moreover, the insight to the structure of the feedback control is partially lost in such a consideration.

The optimal feedback solution is here obtained by means of Fredholm resolvent theory. The resulting equations are more tractable and more readily computable. The optimal feedback controller is compared with the one obtained in [1] and a connection between the two is indicated.

The following notational conventions will be adopted throughout the paper. Let the set S be defined as

$$S = \{t \mid t = t_0 + i,\ i = 1, 2, \ldots\}.$$

The space of R^n-valued continuous functions and differentiable functions with the usual sup topology is denoted by $C([a,b],R^n)$ and $C^1([a,b]/S, R^n)$, respectively. If $x\colon\ [a-1,b] \to R^n$, for some $t>o$, then $x_t(\theta)$, for each $t \varepsilon\ [a,b]$ denotes the elements of $C([a,b],R^n) = C$ given by $x_t(\theta) = x(t+\theta)$, $\theta\ \varepsilon\ [-1,0]$.

FORMULATION OF THE PROBLEM

The state transition of the system is governed by

$$\dot{x}(t) = A_0(t)\, x(t) + A_1(t)x(t-1) + A_2(t)\, \dot{x}(t-1) + B(t)u(t), \text{for } t>t_0, \quad (1)$$

with the intial data

$$x(t) = g(t) \quad \text{on } t \varepsilon [t_0-1, t_0], \quad (2)$$

where $x(t)$ is an n-vector; $u(t)$ is an m-vector; the nxm matrices $A_0(t)$ and $A_1(t)$ are continuous, $A_2(t)$ has a continuous derivative and the nxm matrix $B(t)$ is continuous; the initial function $g(t)$ has a continuous derivative on (t_0-1, t_0). The set of admissible controls to be considered is all piece-wise continuous and square-integrable functions.

The cost functional to be minimized is

$$C(u) = \int_{t_0}^{T} \{x'(s)Q(s)x(s) + u'(s)W(s)u(s)\}\, ds \quad (3)$$

where $Q(s)$ is a continuous, symmetric, positive semidefinite nxn matrix $\forall\, s \,\varepsilon\, [t_0, T]$, and $W(s)$ is a continuous, symmetric, positive definite mxm matrix $\forall\, s \,\varepsilon\, [t_0, T]$, and t_0 and T are given real numbers.

The problem is to minimize the cost functional subject to system constraint (1) by determining the optimal control among the class of admissible control. Moreover, the optimal controller is to be in the feedback form, that is, as a function of the process history.

OPTIMAL FEEDBACK SOLUTION

The maximum principle for the problem is furnished in [9]. It can be stated as

Theorem 1: Suppose $u^*(t)$ is the optimal control and that $x^*(t)$ is the generated optimal trajectory. Then corresponding to $u^*(t)$ and $x^*(t)$, there exists a piece-wise continuous n-vector $p^*(t)$ such that $x^*(t)$ and $p^*(t)$ is a solution of (1), (2), and (4), (5), (6):

$$\dot{p}(t) = -A_0'(t)\, p(t) - A_1'(t+1)\, p(t+1) + \frac{d}{ds}\{A_2'(s)p(s)\}\Big|_{s = t+1}$$

$$-Q(t)x(t) \quad , \quad t \,\varepsilon\, [t_0, T] \quad , \quad (4)$$

$$p(t) = 0 \quad , \quad t \geq T \quad , \quad (5)$$

$$\Delta p(\alpha_i) = A_2(\alpha_i + 1)\, \Delta p(\alpha_i + 1) \;, \; i = 1, \ldots, k + 1, \quad (6)$$

where $\alpha_{k+1} = T$, $\alpha_{i-1} = \alpha_i - 1$ and

$\Delta p(\alpha_i) = p(\alpha_i^-) - p(\alpha_i^+)$, and the optimal control is given by

$$u^*(t) = -W^{-1}(t)\, B'(t)\, p(t). \quad (7)$$

An immediate consequence of the transversality condition $p(T) = 0$ is that $\Delta p(T-1) = 0$ which then implies that $\Delta p(\alpha_i) = 0$ for all $i = 1, \ldots, k+1$. Thus the costate vector $p(t)$ is in fact continuous for all $t \,\varepsilon\, [t_0, T]$.

One can now proceed in the same manner as in [11, 13]. The canonical equations (1) and (4) are converted into integral equations by using the representation of the solution given in Bellman and Cooke [4].

Let the kernel X (t,s) and Y(t,s) be defined as follows:

Definition 1:

(a) Let X(t,r) denote a unique function which satisfies the equation

$$\Gamma X(t,r) = \frac{\partial X(t,r)}{\partial t} - A_0(t)\, X(t,r) - A_1(t)\, X(t-1,r) - A_2(t)\, \frac{\partial X(t-1,r)}{\partial t} = 0\,, \qquad (8)$$

$$t > r,\ t \neq r + n,\ n = 0, 1, 2, \ldots,$$

and the conditions

$$\text{(b)}\quad X(t,r) = 0 \qquad r-1 \leq t < r$$
$$= 1 \qquad t = r \qquad (9)$$

and

(c) $X(t,r) - A_2(t)\, X(t-1,r)$ is a continuous function of t for $t \geq r$.

Definition 2:

(a) Let Y(s,t) denote a unique function with satisfies the equation

$$\Gamma^* Y(s,t) = \frac{\partial Y(s,t)}{\partial s} + A_0'(s)\, Y(s,t) = A_1'(s+1)\, Y(s+1,t) - \frac{\partial}{\partial s}\{A_2'(\alpha)\, Y(\alpha,t)\}_{\alpha = s+1} = 0\,,\quad t_0 < s < t,\ s \neq t-n,\ n=0,1,2,\ldots, \qquad (10)$$

and the conditions

$$\text{(b)}\quad Y(s,t) = 0 \qquad t < s \leq t + 1 \qquad (11)$$
$$= 1 \qquad s = t$$

(c) $Y(s,t) - A_2'(s+1)\, Y(s+1,t)$ is a continuous function of s for $t_0 \leq s \leq t$.

The standard continuation arguments show that X(t,r) and $\frac{\partial X(t,r)}{\partial t}$ are continuous for $t > r$ except for possible jumps at $t = r + n$ $(n + 1, 2, 3, \ldots)$, also, Y (s,t) and $\frac{\partial Y(s,t)}{\partial s}$ are continuous for $t_0 < s < t$, $s \neq t-n$ $(n=0,1,2,\ldots)$.

Assuming that the process history is known up to time τ, that is, both x(s) and $\dot{x}(s)$ are known for all $s \in [t_0-1, \tau]$, be represented as

$$x(t) = [Y'(\tau,t) - Y'(\tau+1,t)\, A_2(\tau+1)]\, x(\tau) + \int_{\tau+1}^{\tau} Y'(s+1,t)\, [A_1(s)\, x(s) + A_2(s+1)\, \dot{x}(s)]\, ds - \int_{\tau}^{t} Y'(s,t)\, B(s)\, W^{-1}(s)\, B'(s)\, p(s)\, ds \qquad (12)$$

and p(t) as

$$p(t) = \int_{t}^{T} X'(s,t)\, Q(s)\, x(s)\, ds, \qquad (13)$$

where equation (2), (5), and (7) have been used.

It is shown in the Appendix that the kernels X(t,s) and Y(s,t) are related as follows:

$$X(t,s) = Y'(s,t) \qquad (13)$$

For convenience of notation, define the matrix M(t,s) by

$$M(t,s) = \int_{\max[t,s]}^{T} X'(r,t)\, Q(r)\, X(r,s)\, dr \tag{14}$$

where the kernel X(t,r) is given above by equation (10). Now using equation (13), the substitution of x(t) from equation (12) into equation (13) results in an integral equation for p(t) of Fredholm type:

$$p(t) = q(t) - \int_{\tau}^{T} K(t,s)\, p(s)\, ds \tag{15}$$

where

$$q(t) = [M(t,\tau) - M(t,\tau+1)\, A_2(\tau+1)]\, x(\tau) + \int_{\tau-1}^{\tau} M(t,s+1)\, [A_1(s)x(s) + A_2(s+1)\, \dot{x}(s)]\, ds, \tag{16}$$

and

$$K(t,s) = M(t,s)\, B(s)\, W^{-1}(s)\, B'(s). \tag{17}$$

It is now easy to demonstrate that the integral equation (15) and the neutral equation (4) exhibit the same solution.

Fredholm's first theorem [15] implies that the solution to equation (15) can be written in the form of equation (18).

$$p(t) = q(t) - \int_{\tau}^{T} R(t,r,\tau)\, q(r)\, dr \tag{18}$$

where the Fredholm resolvent R(· ,· ,·) satisfies the equations

$$R(t,s,\tau) = K(t,s) - \int_{\tau}^{T} R(t,r,\tau)\, K(r,s)\, dr \tag{19}$$

$$= K(t,s) - \int_{\tau}^{T} K(t,r)\, R(r,s,\tau)\, dr \tag{20}$$

For convenience of notation a modified resolvent P(·,·,·) is defined as

$$P(t,s,\tau) = M(t,s) - \int_{\tau}^{T} R(t,r,\tau)\, M(r,s)\, dr \tag{21}$$

It follows from equations (18) and (16) that p(t) can be written as

$$p(t) = [P(t,\tau,\tau) - P(t,\tau+1,\tau)\, A_2(\tau+1)]\, x(\tau) + \int_{\tau-1}^{\tau} P(t,r+1,\tau)\, [A_1(r+1)x(r) + A_2(r+1)\, \dot{x}(r)]\, dr \tag{22}$$

From equation (22) and Theorem 1 the optimal control $u^*(t)$ is then

$$u^*(t) = - W^{-1}(t)\, B'(t)\, \{[P(t,\tau,\tau) - P(t,\tau+1,\tau)A_2(\tau+1)]x(\tau) + \int_{\tau-1}^{\tau} P(t,s+1,\tau)[A_1(s)x(s) + A_2(s+1)\, \dot{x}(s)]ds\, \} \tag{23}$$

where the kernel P(·,·,·) is defined by equation (21).

Integration by parts in equation (23) yields a slightly different form for $u^*(t)$:

$$u^*(t) = - W^{-1}(t)\, B'(t)\, \{P(t,\tau,\tau)[x(\tau) - A_2(\tau)x(\tau-1)] + \int_{\tau-1}^{\tau} [P(t,s+1,\tau)A_1(s) - \frac{\partial}{\partial\alpha}\{P(t,\alpha,\tau)\, A_2(\alpha)\}\Big|_{\alpha=s+1}]x(s)\, ds\, \} \tag{24}$$

By setting $\tau=t$ in equations (23) and (24) the feedback solution is obtained. The latter expression for $u^*(t)$ then corresponds to the one given in Ref. [1]. However, it has to be established that the partial derivative in equation (24) exists.

It is now important to concentrate the attention to the modified resolvent $P(t,s,\tau)$, since the form of the optimal feedback controller has been

INTEGRAL EQUATION FOR $P(t,s,\tau)$

For computational purposes an integral equation for $P(.,.,.)$ is derived. Utilizing the defining equation (21) for $P(.,.,.)$, equations (17) and (19) yield

$$R(t,s,\tau) = P(t,s,\tau)\, B(s)\, W^{-1}(s)\, B'(s) \qquad (25)$$

Substituting equation (25) to equation (21) and using the symmetricity of $M(t,s)$, a Fredholm integral equation for $P(\cdot,\cdot,\cdot)$ is obtained:

$$P(t,s,\tau) = M(t,s) - \int_{\tau}^{T} P(t,r,\tau)\, K'(s,r)\, dr \qquad (26)$$

$$= M(t,s) - \int_{\tau}^{T} K(t,r)\, P(r,s,\tau)\, dr\ . \qquad (27)$$

$P(t,s,\tau)$ can now be computed from equation (26) or (27) by standard numerical methods used in solving Fredholm integral equations. Of course, if the interval $[t_0,T]$ is large this is not an easy task, especially, since the domain of $P(t,s,\tau)$ is 3-dimensional $\{(t,s,\tau) | t,s\geq\tau\ ,\ T\geq\tau\geq t_0\}$. However, it appears to be more promising computationally than the result derived in [1], where several partial differential equations have to be solved on each unit interval.

One can also derive partial differential equations (of neutral type) for $P(\cdot,\cdot,\cdot)$ starting from equation (27) in the similar fashion as in [11].

CONCLUSION

It has been demonstrated that the powerful Fredholm resolvent theory can solve the difficult split boundary value problem arising in optimizing a linear neutral system with a quadratic cost. The feedback law depends on the state and also on the history of the state and its derivative.

A Fredholm integral equation that appears to be computationally more feasible than the partial differential equations of Ref. [1] is derived for the optimal gain matrix $P(\cdot,\cdot,\cdot)$.

APPENDIX

Consider the following identity

$$0 = \int_{r}^{t} \{ Y'(\sigma,t)\ \ \Gamma X(\sigma,r) - (\Gamma^{*} Y(\sigma,t))'\, X(\sigma,r)\}\, d\sigma\ \ .$$

Utilizing the definitions of $\Gamma X(t,r)$ and $\Gamma^{*}Y(s,t)$ and the properties of $X(t,r)$ and $Y(s,t)$ one obtains from the above identity

$$0 = \int_{r}^{t} \frac{\partial}{\partial\sigma} \{ Y'(\sigma,t)\ [X(\sigma,r) - A_2(\sigma)\ X(\sigma-1,r)]\}\, d\sigma$$

From the second identity one has

$$Y'(r,t) = X(t,r) - A_2(t)\, X(t-1,r) + \sum_{j=1}^{M} [\, Y'(t-j-,t) - Y'(t-j+,t)]\, [\, X(t-j,r) - A_2(t-j) X(t-j,r)]$$

or

$$Y'(r,t) = X(t,r) - A_2(t)\, X(t-1,r) + \sum_{j=1}^{M} \prod_{k=1}^{j} A_2(t-k-1)\, [X(t-j,r) - A_2(t-j)\, X(t-j-1,r)] ,$$

where M is the largest integer such that $r < t-M$. From this expression for $Y(r,t)$ one can now easily derive that

$$Y'(r,t) - A_2(t)\, Y'(r,t-1) = X(t,r) - A_2(t)\, X(t-1,r).$$

The result $Y'(r,t) = X(t,r)$ follows by first considering the interval $r \leq t < r+1$ and then inductively.

REFERENCES

[1] J.K. Aggarwal and D.H. Eller,"Optimal control of neutral systems", Int. J. on Control, Vol. 14, No.2, 1971, 308-319.

[2] Y. Alekal, P. Brunovsky, D.H. Chyung, and E.B. Lee," The quadratic problem for systems with time delays", IEEE Tr. on Automatic Control, AC-15,1971, 673-678.

[3] H.T. Banks and G.A. Kent,"Control of functional differential equations to target sets in function space", SIAM J. on Control, Vol. 10,1972, to appear.

[4] R.L. Bellman and K.L. Cooke, Differential-Difference Equations, Academic Press, N.Y., 1963.

[5] R.K. Brayton,"Nonlinear oscillations in a distributed network", Quart. Appl. Math.,Vol. 24,1967, 289-301.

[6] R. Driver,"A functional-differential equation arising in a two body problem of electrodynamics", Internat. Symp. of Nonlinear Differential Equations and nonlinear Mechanics, Academic Press, 1963, 474-484.

[7] R. Driver, "Existence and continuous dependence of solutions of a neutral functional-differential equation", Archives for Rational Mechanics, Vol. 19, 1965, 149-166.

[8] J.K. Hale and K. Meyer," A class of functional equations of neutral type", Memoirs of American Mathematical Society, No.76,1967.

[9] G.A. Kamenskii and E.A. Khvilon,"Necessary conditions for the optimal control of systems with deviating argument of neutral type", Automation and Remote Control, Vol.30,1969,327-339.

[10] H.N.Koivo,"A survey of optimal control of hereditary systems", Applied Mechanics Reviews, Vol. 25, 1972, 239-244.

[11] H.N. Koivo and E.B. Lee,"Controller synthesis for linear systems with retarded state and control variables and quadratic cost", Automatica, Vol.8, 1972, 203-208.

[12] A. Manitius,"Optimal control of processes with time-delays - - a survey and some new results", Arch. Aut. i Telemech., Vol.15, 1970,205-221.

[13] A. Manitius,"Optimum control of linear time lag processes with quadratic performance indexes", The Fourth IFAC Congress, Warsaw, Poland,1969.

[14] L.W. Neustadt," A general theory of extremals", J. Computer and system Science, Vol. 3, 1969, 57-92.

[15] W. Pogorzelski, Integral Equations and Their Applications, Pergamon Press, Warsaw,1969.

[16] V.P. Rubanik,"Oscillations of quasilinear systems wth retardation", Nauka, Moscow, 1969, (Russian).

MODELING A SET OF SEQUENTIAL DATA[†] [‡]

JI-GUAN LIN
Department of Electrical Engineering and Computer Science
Columbia University
New York, New York 10027

ABSTRACT:

The modeling of ℓ sequential data of length k into linear autonomous discrete (LAD) systems of the minimal order is considered first, then its various applications to reduction of partially observable or partially controllable systems (discrete or continuous), to identification of LAD systems.

A necessary and sufficient condition under which the sequential data have an nth-order LAD model is given. It is shown that the minimal order N is given by the sum of two integers n_r and β. A modeling algorithm is given for determining N and a minimal-order LAD model. Examples are illustrated as results of the use of a modeling program developed according to the algorithm.

1. INTRODUCTION

Consider ℓ sequential data of length k $(1 \le \ell < \infty,\ 1 \le k < \infty)$ given by

$$y^i = (y^i(1), y^i(2), \ldots, y^i(k)), \quad i = 1, 2, \ldots, \ell . \tag{1.1}$$

These sequences may be the available observation data of a system in a "black box", or the daily measurements of air quality, or the deviations (from the normal) of the patient's temperature readings. In all these cases the inputs to or the initial disturbances on the system from which the data derive may be unknown, so is the system itself. In order to be able to analyze, to control, or to improve the system, or to predict future data, or to realize the current state of the system, etc., one would try either to find curves that best fit the data (in the sense of the least squared error, for instance) or to look for a rather simple, convenient model that takes the data as its output sequences. Systems scientists would generally desire models, if any, as simple as (deterministic) linear autonomous (i. e. time-invariant) systems represented by a system of linear difference (or

---

[†] This research was mainly supported by the National Science Foundation under Grant NSF-GK-32701, and partially by Columbia University Pool Funds for computer services.

[‡] For the Tenth Annual Allerton Conference on Circuit and Systems Theory, University of Illinois; October 4-6, 1972.

differential) equations with constant coefficients of the minimal order.

In this paper a necessary and sufficient condition under which the data have an nth-order LAD (linear autonomous discrete) model will be given first (Theorem 2.1). One might conjecture that the rank r of the data matrix is the minimal order; it will be illustrated, however, that the minimal order N is generally higher than r. It will be shown that N is given by the sum of two integers n_r and β (Theorem 2.3). A modeling algorithm for determining N and a minimal-order LAD model will be presented (Theorem 3.1). A program can be developed using this algorithm with minimal requirement on storage. Some examples will be illustrated as the results of the use of a modeling program.

Applications of the modeling algorithm and technique will be made to the reduction of partially observable, or partially controllable linear autonomous systems, discrete or continuous; application is also made to the identification of LAD systems. All seems to be easy once a modeling program is developed according to the modeling algorithm. We shall notice the relative easiness of application in the processes of reduction and of identification.

2. EXISTENCE OF nth-ORDER LINEAR AUTONOMOUS MODELS FOR THE DATA, AND THE MINIMAL ORDER N.

We shall for the time being regard the data set (1.1) as the ℓ sequential observations of a single-output system, for the extension to multiple-output cases is rather straightforward then. We shall also assume that $\ell \le k$. Otherwise, consider those k sequences of which the remaining $(\ell - k)$ are linear combinations.

The data set (1.1) is said to have an nth-order LAD model (linear autonomous discrete model) if there exist an $n \times n$ real constant matrix A, a real constant n-vector c, and ℓ real n-vectors $x^1, x^2, \ldots, x^\ell$ such that each sequence in set (1.1), say y^i, corresponds to the output sequence of length k starting in an initial state $x(1) = x^i$ of the following system

$$\begin{cases} x(t+1) = Ax(t) \\ y(t) = c^T x(t) \qquad t = 1, 2, \ldots \end{cases} \tag{2.1}$$

where T denotes the transpose. $x(t)$ is called the state, and $y(t)$ the output, at time t. Two LAD models $\{A, c^T, x^1, x^2, \ldots, x^\ell\}$ and $\{B, d^T, z^1, z^2, \ldots, z^\ell\}$, not necessarily of the same order, are said to be equivalent for the data set if the outputs of both models coincide with the data set.

Suppose n is a given integer. Let the row n-vector $[y^i(j)\ y^i(j+1) \ldots y^i(n+j-1)]$ be simply denoted by $y^i(j, n+j-1)$. We can check the existence of an nth-order LAD model for the data (1. 1) by the following condition (2. 2).

Theorem 2. 1 [1]

(a) The data set (1. 1) has an LAD model of order $n (n < k)$ if and only if there exists a real n-vector p with which the following condition is satisfied:

$$y^i(n+j) = y^i(j, n+j-1)p \quad j = 1, 2, \ldots, k-n; \ i = 1, 2, \ldots, \ell. \qquad (2.2)$$

(b) The data set (1. 1) always has an LAD model of order k.

Concluding from the sufficienty proof of the above theorem, we have $\{F, e_1^T, y^1(1, n)^T, y^2(1, n)^T, \ldots, y^\ell(1, n)^T\}$ as an nth-order LAD model for the data set (1. 1) if it satisfies condition (2. 2) for some real n-vector p. F is given by

$$\begin{bmatrix} 0 & 1 & & & \bigcirc \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ \bigcirc & & 0 & 1 & 0 \\ & & & 0 & 1 \\ p_1 & p_2 & \cdots & & p_n \end{bmatrix}, \text{ and } e_1 \text{ is } \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Although the data (1. 1) can always have a linear autonomous model of order k (see Theorem 2. 1(b)), such a high-order system is not practically desirable in general when k is very large (5000 data points for instance). On the other hand, a model of the minimal order, if obtainable, would be rather desired. One might conjecture that the rank r of the data matrix

$$Y \triangleq \begin{bmatrix} y^1(1) & y^1(2) & \ldots & y^1(k) \\ \vdots & \vdots & & \vdots \\ y^\ell(1) & y^\ell(2) & \ldots & y^\ell(k) \end{bmatrix}$$

is the minimal order; it will be shown, however, that the minimal order N is generally higher than r. See the following Example 2. 1 for illustration.

Example 2.1

$$\begin{aligned} y^1 &= (0,\ 0,\ 1,\ -1,\ 0,\ 0,\ 0,\ 1,\ -1,\ 0) \\ y^2 &= (0,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 1) \\ y^3 &= (0,\ 0,\ -1,\ 1,\ 0,\ 0,\ 0,\ -1,\ 1,\ 0) \\ y^4 &= (0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 1) \\ y^5 &= (0,\ 0,\ 1,\ 1,\ 2,\ 0,\ 0,\ 1,\ 1,\ 2)\ . \end{aligned}$$

The data matrix has rank $r = 2$, for

$$y^5 = y^1 + y^2 + y^3 + y^4, \quad y^4 = y^1 + y^2\ , \quad y^3 = -y^1\ .$$

Condition (2.2) is not satisfied for $n \le 4$, but is for $n = 5$, with $p^T = [1\,0\,0\,0\,0]$. $\therefore N = 5,\ r = 2$.

The task now is to find the minimal order for the data. To start with, denote by n_r the smallest n $(1 \le n \le k)$ for which the $\ell \times n$ matrix

$$\begin{bmatrix} y^1(1,n) \\ y^2(1,n) \\ \vdots \\ y^\ell(1,n) \end{bmatrix} \text{ has the rank } r \text{ of the data matrix } \begin{bmatrix} y^1(1,k) \\ y^2(1,k) \\ \vdots \\ y^\ell(1,k) \end{bmatrix}\ .$$

N is said to be the minimal order for the data (1.1) if N is the lowest order of all LAD models equivalent for the data.

Theorem 2.2 [1, 2]

$$N \ge n_r \ge r.$$

In fact, there is some difference between N and n_r. A sketch of proof for Theorem 2.3 below is given, since the procedure involved is useful for developing a modeling algorithm.

Theorem 2.3 [1]

There is an integer β, not necessarily zero, such that $n_r + \beta = N$.

Proof (sketch):

Suppose that condition (2.2) in Theorem 2.1 is successively satisfied for all j up to some integer, say, j_n depending on n. If $j_n < k-n$, the data set (1.1) has no LAD models of order n. Define $k_n = n + j_n$. k_n is the length for which the data set has an nth-order LAD model.

Claim (a) The sequence $\{k_n\}_{n=n_r}^{n=k}$ is nondecreasing, and

(b) For each n' with $k_{n'} < k$, there is $n > n'$ such that $k_n > k_{n'}$.

Properties (a) and (b) altogether mean that $\{k_n\}$ converges to k at some n^*, which is then shown to be N. In other words, the increase of n to n^* results in increase of the length k_n to k. The times of increase from n_r to n^* is the integer β. $\therefore N = n^* = n_r + \beta$.

In Example 2.1, $n_r = 4$, but $N = 5$. $\therefore$ $\underline{\beta = 1}$.

Example 2.2

$$y^1 = (1, -1, 0, 0, 0, 1, -1, 0, 0, 0)$$
$$y^2 = (0, 1, 1, 0, 0, 0, 1, 1, 0, 0)$$

The data matrix has rank $r = 2$ and $n_r = 2$. By checking the condition (2.2), one can see that $\underline{\beta = 3}$, and $N = 5$.

3. ALGORITHM FOR THE MINIMAL ORDER AND A MINIMAL-ORDER LAD MODEL

We can summarize the above results into an algorithm with which one can find the minimal order N and an LAD model of minimal order for the data. Before proceeding, we shall notice the following fact (Lemma 3.1) on the condition (2.2), when aligning $y^i(j, n+j-1)$ of (2.2) into a system of $\ell(k-n)$ simultaneous equations in n variables.

Lemma 3.1

Condition (2.2) is satisfied if and only if both matrices, the $\ell(k-n)\times n$ matrix Y_n and $\ell(k-n)\times(n+1)$ Y_{n+1}, defined below have the same rank:

$$Y_n \triangleq \begin{bmatrix} Y(1, n) \\ \vdots \\ Y(j, n+j-1) \\ \vdots \\ Y(k-n, k-1) \end{bmatrix}, \quad Y_{n+1} \triangleq \begin{bmatrix} Y(1, n+1) \\ \vdots \\ Y(j, n+j) \\ \vdots \\ Y(k-n, k) \end{bmatrix}$$

where $Y(j, j') \triangleq \begin{bmatrix} y^1(j, j') \\ y^2(j, j') \\ \vdots \\ y^\ell(j, j') \end{bmatrix}$, $1 \le j \le j' \le k$.

Notice that if Lemma 3.1 is applied directly on computers, storage of tremendous size is required; the dimension of the array Y_{n+1} would be at least $\frac{1}{4}k^2$ by $\frac{1}{2}(k+2)$ for all possible n. The following algorithm requires only the minimal storage.

Theorem 3.1 (Modeling Algorithm)

The minimal order N and an LAD model of the minimal order for the data (1.1) can be found by the following steps:

Step 1. Find the rank r and the integer n_r of the data matrix.

Step 2. For the integer n (starting with $n = n_r$), collect first n (linearly) independent rows for the matrix Y_{n+1}, hence for Y_n, by aligning $y^i(j, n+j-1)$ starting with $j = 2$ and $i = 1, 2, \ldots, \ell$ for each j. Whenever inconsistency occurs, that is, whenever both Y_{n+1} and Y_n so collected have different ranks, restart Step 2 with n increased by 1. If at the end the number of consistent rows is less than n, jump to Step 4.

Step 3. Check one by one if each of the remaining row vectors $y^i(j, n+j-1)$ is a linear combination of the first n independent rows collected in Step 2. If not, go back to Step 2 with n increased by 1.

Step 4. $\beta = n - n_r$, the minimal order $N = n_r + \beta$.

Step 5. Solve the N-vector p using the independent equations (i.e. rows of Y_{n+1}) collected in Step 2. (Notice that the number of rows is at most N.)

Step 6. $F^T = [e_2\, e_3 \ldots e_N p]$. A minimal-order LAD model for the data is $\{F,\ e_1^T,\ y^1(1, N)^T,\ y^2(1, n)^T,\ \ldots,\ y^{\ell}(1, N)^T\}$.

A modeling program has been developed to implement the modeling algorithm. The subroutine MFGR in the IBM Scientific Subroutine Package is called to find ranks of matrices. Examples 2.1 and 2.2 were then solved by this modeling program.

4. APPLICATIONS TO SYSTEM REDUCTION

Besides the use in modeling itself, the modeling algorithm can be employed to reduce linear autonomous systems, discrete or continuous, from partially observable to completely observable, or from partially controllable to completely controllable. We shall notice the relative easiness in the process of reduction.

Reducing partially observable discrete systems will be considered first, since extension to other problems will become obvious then. The invariance of r, n_r and N under nonsingular transformations is the key factor.

Theorem 4.1

The integers r, n_r and N pertaining to the ℓ sequential data (1.1) are invariant under $\ell \times \ell$ nonsingular transformations on the data.

1. Partially Observable Discrete Systems

Suppose an nth-order LAD system is given by (2.1). Pick (arbitrarily) n linearly independent initial states $x^1, x^2, \ldots, x^n$. With the help of (2.3) we can write the corresponding output sequence $y^1, y^2, \ldots, y^n$ as

$$\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix} = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^n)^T \end{bmatrix} [c \ A^T c \ldots (A^T)^{k-1} c] .$$

Take these output sequences as the sequential data (where $\ell = n$), transform them by the $n \times n$ nonsingular matrix

$$T \triangleq \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^n)^T \end{bmatrix}^{-1} ,$$

and get $[c \ A^T c \ldots (A^T)^{k-1} c]$ as the transformed data matrix. By Theorem 4.1 we can apply the modeling algorithm directly on the transformed sequential data and find the minimal order N and the N-vector p for an equivalent minimal-order system $\{F, e_1{}^T\}$. Once the algorithm is implemented by a computer program, a reduced (completely observable) system can be found easily.

In fact, by observability theory, it is sufficient to consider the transformed sequential data for length k equal to the order n of the reducible system, or only equal to the rank r of the observability matrix $[c \ A^T c \ldots (A^T)^{n-1} c]$, once it is found.

Example 4.1

Consider an LAD system $\{A, c^T\}$ given by

$$A^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & -5 \end{bmatrix}, \quad c = \begin{bmatrix} -1 \\ 2 \\ 1 \\ 3 \\ 0 \end{bmatrix}.$$

By the modeling program mentioned previously the minimal order N is found to be 3 and $p = \begin{bmatrix} 0 \\ -4 \\ -4 \end{bmatrix}$. Thus $\left\langle \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & -4 \end{bmatrix}, [1\,0\,0] \right\rangle$ is a desired minimal-order LAD system equivalent to $\{A, c^T\}$.

2. Partially Observable Continuous Systems

Suppose that an nth-order linear autonomous continuous system is given by

$$\dot{x}(t) = A\,x(t)$$

$$y(t) = c^T x(t)$$

It is well-known that the observability matrix of this system is $[c\ A^T c \ldots (A^T)^{n-1} c]$, which is exactly in the same form as the transformed sequence data of length n in the case of discrete systems. Therefore the modeling algorithm or the modeling program can again be directly applied on the sequential data $[c\ A^T c \ldots (A^T)^{n-1} c]$ to find the vector p, hence the matrix F for a minimal-order (completely observable) linear autonomous continuous system.

3. Partially Controllable Systems, Discrete or Continuous

Let an nth-order linear autonomous system be given by

$$x(t+1) = A\,x(t) + b\,u(t) \qquad t = 1, 2, \ldots$$

or by

$$\dot{x}(t) = A\,x(t) + b\,u(t) \qquad t \geq 0$$

where b is a real n-vector and u(t) is scalar input. It is well known that the controllability matrix of either system is $[b\ Ab \ldots A^{n-1} b]$. Comparing with cases (1) and (2) above, one can as well apply again the modeling algorithm on these sequential data and find a corresponding reduced (completely controllable) linear autonomous system.

5. APPLICATION TO SYSTEM IDENTIFICATION

Suppose that λ n-vectors, say $x^1, x^2, \ldots, x^\lambda$, are conjectured to be a set of the initial states of the unknown LAD system which puts out the sequential data (1. 1), and the integer n is to be the order. One wants to identify, if possible, this unknown system. We shall qualify these conjectures first (by Theorem 5. 1 below), then identify one n th-order LAD system from those which are equivalent to the minimal-order system found by the modeling algorithm of Theorem 3. 1 (Theorem 5. 2).

Denote by ρ the rank of the conjectured initial-state matrix $[x^1\ x^2\ \ldots\ x^\lambda]$. Recall the rank r of the data matrix. As before, let N be the minimal order and $\{F, e_1^T, y^1(1,N)^T, y^2(1,N)^T, \ldots, y^\ell(1,N)^T\}$ be the minimal-order model for the data (1, 1) found by the modeling algorithm or the modeling program .

Theorem 5. 1

If there is an n th-order LAD system having $x^1, x^2, \ldots, x^\lambda$ as initial states and the sequential data (1. 1) as output sequences, it is necessary that (a) $\lambda \le \ell$, (b) $n \ge N$, (c) $\rho \ge r$, and

$$\text{(d)}\quad \operatorname{rank}\left(\left[\begin{array}{c|c} (x^1)^T & y^1(j) \\ (x^2)^T & y^2(j) \\ \vdots & \vdots \\ (x^\lambda)^T & y^\lambda(j) \end{array}\right]\right) = \rho, \quad \text{for all } j = 1, 2, \ldots, N .$$

Suppose now that conditions (a) - (d) are satisfied. Then there exist N n-vectors $v_1, v_2, \ldots, v_N$ such that

$$\begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^\lambda)^T \end{bmatrix} v_j = \begin{bmatrix} y^1(j) \\ y^2(j) \\ \vdots \\ y^\lambda(j) \end{bmatrix}, \quad j = 1, 2, \ldots, N . \tag{5. 1}$$

Set $V \triangleq [v_1\ v_2 \ldots v_N]^T$. Adding some appropriately chosen n-vectors $x^{\lambda+1}, x^{\lambda+2}, \ldots, x^\ell$ if $\ell > \lambda$, we have from (5. 1)

$$V x^i = y^i(1,N)^T \qquad i = 1, 2, \ldots, \ell \tag{5. 2}$$

Theorem 5.2

Suppose the conditions (a) - (d) in Theorem 5.1 hold and also that $r = N$. Then the right inverse V^R of the $N \times n$ matrix exists, and an nth-order LAD system, say $\{A, c^T\}$, having $x^1, x^2, \ldots, x^\lambda$ as initial states and the sequential data (1.1) as output sequences, is given by

$$A \triangleq V^R F V, \quad c^T \triangleq e_1^T V \quad .$$

Applications to other problems (approximation of nonautonomous linear systems for example) are possible and extension to multiple-output systems does not seem to be too difficult [1].

REFERENCES

[1] Lin, J. G., "Modeling of Sequential Data", Tech. Report, Systems Research Group, Department of Electrical Engineering and Computer Science, Columbia University, September 1972.

[2] Lin, J. G., "Modeling of Linear Systems from Given Sets of Sequences," Tech. Report, Systems Research Group, Department of Electrical Engineering and Computer Science, Columbia University, May 1970.

[3] Gopinath, B., "On the Identification of Linear Time-Invariant Systems from Input-Output Data," Bell System Tech. J., vol. 48, no. 5, pp. 1101-1113, May 1969.

[4] Cohn, M. and Even, S., "Identification and Minimization of Linear Machines," IEEE Trans. on Electronic Computers, vol. EC-14, pp. 367-376, June 1965.

[5] Zadeh, L. A. and Desoer, C. A., Linear System Theory: The State Space Approach, McGraw-Hill, N.Y. 1963.

[6] Sarachik, P. E. and Kreindler, E., "Controllability and Observability of Linear Discrete-Time Systems," International J. Control, vol. 1, no. 5, pp. 419-432, 1965.

ON GRAPH THEORY AND CLUSTER ANALYSIS[†]

Zen Chen and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

In this paper several concepts of graph theory which are useful in cluster analysis are formally introduced. Through the use of graph theory a better understanding of the clustering problem can be obtained, and some of the serious difficulties encountered in cluster analysis can be overcome by graph-theoretical methods.

I. INTRODUCTION

Given a collection of objects, each described by the same set of attributes, a clustering problem is, roughly speaking, to classify these objects, based on the available descriptions of objects, into several groups such that members of each group are similar to each other. A clustering technique is mainly a two-stage process. First, there is a selection of pairwise similarity measure between every two objects. Secondly, the chosen similarity measure is incorporated into a sorting strategy whereby groups of similar objects are extracted.

Several concepts of graph theory have been shown to be useful in cluster analysis. Within the framework of graph theory, clusters obtained by current clustering techniques can be interpreted as different types of graphs such as components, complete subgraphs, cliques, and maximal connected subgraphs of a certain minimum degree, etc. In an advanced study of graph-theoretic methods, the notion of the connectivity of graphs plays a central role in the investigation of the data structure of clusters. The troublesome chaining effect of a cluster which results in misclassification in pattern recognition is reflected by the existence of small articulation sets. This serious problem can be formulated as a network flow problem of multiple sources and multiple sinks. An algorithm is developed for removing the small articulation sets and producing sequential subclusters according to a prespecified splitting criterion.

The graph-theoretical approach to cluster analysis has several advantages over the other existing clustering techniques. For example, clusters obtained by the graph-theoretical approach have a clear, explicit interpretation, different from the fuzzy notion of clusters obtained by most of the other methods. However, current graph-theoretical methods have weaknesses, and some improvements can be made. A complete investigation of graph-theoretical methods for cluster analysis is still required.

II. GRAPH-THEORETICAL METHODS FOR CLUSTER ANALYSIS

Graph-theoretical methods for cluster analysis have been studied in [1-10]. Before a brief summary of these existing results is made, several important terminologies in graph theory are first introduced [11-13].

Definition 1: A graph $G = (V,\Gamma)$ consists of a finite nonempty set V of p distinct elements, called vertices, and a set of q unordered pairs of

---

[†] This work was supported by National Science Foundation Grant GK-18225.

distinct vertices of V, called edges. Two vertices linked by an edge are called adjacent vertices, and each vertex and the edge are incident with each other.

Definition 2: A subgraph $G' = (V',\Gamma')$ of a graph $G = (V,\Gamma)$ is a graph with $V' \subseteq V$ and $\Gamma' \subseteq \Gamma$. A trivial subgraph of G is a graph containing one vertex.

Definition 3: A path in a graph G is a sequence of distinct vertices and distinct edges $v_o,e_1,v_1,\ldots,v_{n-1},e_n,v_n$ where e_i is the edge between v_{i-1} and v_i, $i=1,2,\ldots,n$. If $v_o = v_n$, then the sequence $v_o,e_1,v_1,\ldots,v_{n-1},e_n,v_n$ is a circuit of G.

Definition 4: A graph G is connected if every pair of vertices in G are joined by a path. A graph is disconnected if it is not connected.

Definition 5: A connected subgraph of a graph G is called a component of G if it is not contained in any other connected subgraph of G.

Definition 6: The degree of a vertex v in a graph G is the number of edges incident with v. The minimum (maximum) degree of G is the smallest (largest) degree that a vertex in G has.

Definition 7: A graph G is said to be regular of degree k if every vertex in G is of degree k.

Definition 8: A graph G is said to be complete if every pair of vertices in G are adjacent. That is, G is complete if G is regular of degree (p-1) where p is the total number of vertices in G.

Definition 9: A complete subgraph of a graph G is called a clique of G if it is not contained in any other complete subgraph of G.

Definition 10: A tree of a graph G is a connected subgraph which contains no circuits. A spanning tree of G is a tree which contains all vertices of G. If each edge of G is associated with a number, called the weight, a minimum spanning tree of G is a spanning tree which has the minimum sum of weights of all edges in the tree.

In a clustering problem, let the given collection of objects be denoted as $X = \{x_1,x_2,\ldots,x_n\}$ and let the pairwise similarity measure between every two objects, x_i and x_j, in X be denoted as s_{ij}, $i \neq j$, $i,j=1,2,\ldots,n$. Then the clustering problem can be formulated within the framework of graph theory in the following way. Consider a graph $G = (V,\Gamma)$ with $V = X$ and $\Gamma = \{(x_i,x_j) \mid x_i \neq x_j,\ x_i \in X,\ x_j \in X$ and s_{ij} satisfies certain conditions$\}$. A graph-theoretical clustering technique is basically a method of specifying the conditions for the construction of edges and the subsequent mechanism of searching for all subgraphs of a certain kind in G.

The application of graph theory to cluster analysis began about 1960 [1]. Various kinds of graphs are defined as clusters in existing graph-theoretical approaches to cluster analysis . Cliques are defined as clusters in [1,2], components in [3,4]. In [5] Ling defined a (k,r)-cluster as a maximal connected graph of minimum degree k. In [6-8] Jardine, et.al., defined an ultimate cluster at each level obtained by the k-partition clustering procedure as a connected graph of a minimum degree greater than a certain number such that the maximum number of objects in the overlaps between clusters is (k-1). The cluster formed in the k-linkage algorithm

[9] is a connected subgraph of minimum degree not less than k or a complete subgraph when the number of group members is less than k. Recently, Zahn [10] proposed that clusters can be obtained through the minimal spanning tree methods.

III. THE ROLE OF CONNECTIVITY

Existing graph-theoretical methods for cluster analysis have not thoroughly investigated the data structures of clusters. The connectivity of clusters will be studied in some detail in this section.

Definition 11: A vertex cut-set of a connected graph G is a set of vertices whose removal (including all edges incident to vertices in the set) from G leads to a disconnected or trivial subgraph. An S-T vertex cut-set is a set of vertices (excluding vertices in S and T) whose removal from G interrupts all paths between two subsets of vertices S and T (S and/or T may contain only one vertex). An edge cut-set of a connected graph can be defined similarly. A vertex cut-set is also called an articulation set.

Definition 12: The connection number (or the vertex-connectivity) of a connected graph G is the number of vertices in a minimal vertex cut-set. The cohesion number (or the edge-connectivity) of a connected graph G is the number of edges in a minimal edge cut-set.

Theorem (Whitney): In any graph G, the minimum degree $\delta(G)$, the connection number $\omega(G)$, and the cohesion number $\chi(G)$ satisfy the following inequality:

$$\omega(G) \leq \chi(G) \leq \delta(G) .$$

A proof of this theorem can be found in [11].

Definition 13: If h is an integer ≥ 1, G is said to be h-connected if $\omega(G) \geq h$, or G is said to be h-coherent if $\chi(G) \geq h$.

Corollary: If a graph G is h-connected or h-coherent, where h is some positive integer, then the degree of every vertex in G is greater than or equal to h.

Notice that if the minimum degree of a graph G is h, it is not necessarily true that G is h-connected or h-coherent.

Theorem: The necessary and sufficient condition for a graph $G = (V,\Gamma)$ to be h-connected (h-coherent) is that two arbitrary vertices $v_i, v_j \in V$ can be joined by h paths which have no vertices (edges) in common except the vertices v_i and v_j.

The proof of the theorem is based on the network flow theory. For a detailed proof, refer to [12].

The troublesome chaining effect can be explained by using the concept of h-connectedness. In Figure 1(a) a cluster containing 10 vertices is a complete graph of degree 3. The chaining effect of the cluster is due to the fact that the cluster is 1-connected instead of being 3-connected.

If a cluster has a high value of h-connectedness, it is tightly connected. However, if a cluster has a low value of h-connectedness, it is not always loosely connected. For example, in Figure 1(b), the cluster is 1-connected, while it is rather tightly connected; the connectivity of the cluster is ruined by a few vertices of degree 1.

A proper way to examine the connectivity of a cluster is to investigate whether there are subgraphs of the cluster, called subclusters, satisfying the following splitting criterion:

(1) the sizes of the vertex cut-sets are small;

(2) the sizes of the resultant subgraphs, after the removal of the vertex cut-sets, are resonably large, and the subgraphs are connected.

In the case where a desirable vertex cut-set is equal to an s-t vertex cut-set between two certain vertices s and t, by the well-known max-flow min-cut theorem, the vertex cut-set can be found by a process to maximize the flow from s to t.

In a cluster viewed as a connected graph $G = (V,\Gamma)$, for two fixed vertices s and t, called the source and the sink, a flow function $f\colon V \times V \rightarrow \{0,1\}$ is subject to

$$
\begin{aligned}
&f(s,V) - f(V,s) = c\\
&f(x,V) - f(V,x) = 0, \quad x \neq s,t, \quad x \in V\\
&f(t,V) - f(V,t) = -c\\
&f(x,V) + f(V,x) \leq 2, \quad x \neq s,t, \quad x \in V\\
&f(x,y) = f(y,x) = 0, \quad \text{if } x \text{ and } y \text{ are not adjacent}\\
&f(x,x) = 0, \quad x \in V
\end{aligned}
$$

where

$$f(y,V) \overset{\Delta}{=} \sum_{z \in V} f(y,z) \quad \text{for some } y \in V$$

and c is a positive integer.

There are many methods to find the maximum flow [12,14-16]. Among them, Frisch's labeling algorithm is one of the most efficient methods. However, there are cases where the size of a desirable vertex cut-set is larger than any minimal s-t vertex cut-set between any two vertices s and t. Therefore, a flow problem of single source and single sink needs to be generalized to a flow problem of multiple sources and multiple sinks. A generalized version of Frisch's labeling algorithm has been described in [17]. To eliminate the chaining effect of a cluster, an algorithm containing the generalized Frisch's labeling algorithm was proposed in [17]. The generalized Frisch's labeling algorithm and the algorithm for eliminating the chaining effect of a cluster are summarized in the appendix.

IV. AN ILLUSTRATIVE EXAMPLE

An example is presented to show the importance of the connectivity investigation and to illustrate the procedure of eliminating the chaining effect.

Example: The dissimilarity coefficients on a collection of objects $V = \{1,2,\ldots,10\}$, $\{d_{ij} \mid i \neq j,\ i,j \in V\}$, are given as follows:

	2	3	4	5	6	7	8	9	10
1	1	1.4	1	2	3	4	3.8	4.8	5.2
2		1	1.4	2	3	3.8	4	5.2	4.8
3			1	1	2	2.8	3	4.1	3.9
4				1	2	3	2.8	3.9	4.1
5					1	2	2	3.1	3.1
6						1	1	2	2
7							1	1.4	1
8								1	1.4
9									1

One way to construct the edges between the objects of V is to obtain a threshold matrix from the table of the dissimilarity coefficients. That is, an edge exists between two objects i and j, $i,j \in V$, if $d_{ij} \leq T$ where T is a preset threshold. A reasonable value of T is 1.5; the corresponding graph for $T = 1.5$ is shown in Figure 2(a). Notice that the graph remains the same for $T \in [1.4,2)$. The graph is connected and the minimum degree of the graph is 3. By most of the existing graph-theoretical clustering techniques, the graph constructed is considered as a cluster. However, through the investigation of the connectivity of the graph only a flow path exists between vertices 1 and 10, and vertex 5 or 6 is the articulation set. The process of detecting the articulation set is shown in Figure 2(b) and (c). The splitting criterion is specified by $k=2$, $m=4$. The source is vertex 1 and the sink is vertex 10. The result of the first labeling iteration is shown in Figure 2(b). The solid arrowheads indicate a newly established flow path. Figure 2(c) shows the final result of the labeling process. The dashed arrowheads indicate the old flow path formed previously. The labeling process terminates because no new label can be assigned. Vertex 5 is the vertex adjacent to an unlabelled vertex (i.e., vertex 6), so it is the articulation set. This investigation of connectivity reveals that there is a chaining effect in the cluster. The parent clusters may be broken into two subclusters to eliminate the chaining effect.

V. CONCLUSIONS AND FURTHER REMARKS

Graph theory has been shown to be a useful tool for cluster analysis. The graph-theoretical methods for cluster analysis have advantages over the other clustering techniques. The notion of connectivity of a cluster plays an important role in the investigation of data structure of a cluster. It has been shown that the chaining effect of a cluster is a consequence of poor connectivity. An algorithm containing the network flow technique (i.e., the generalized Frisch's labeling algorithm) is used to eliminate the chaining effect. Currently an improved graph-theoretical method is under development. It is believed that this method will fulfill many desirable requirements which the existing methods fail to meet.

APPENDIX

A. The Generalized Frisch's Labeling Algorithm (GFLA)

Let a cluster be denoted as $G = (V,\Gamma)$ and let a set of sources be $S = (s_1,\ldots,s_p)$ and a set of sinks $T = (t_1,\ldots,t_q)$, where p and q are two positive integers. A generalized version of Frisch's labeling algorithm is given below. The algorithm assumes a zero initial flow.

Routine A - Labeling Process

1. Label $s_1,\ldots,s_p$ by $(s_1,+),\ldots,(s_p,+)$. Now $S = (s_1,\ldots,s_p)$ is + labeled

and + unscanned. All other vertices are unlabeled and unscanned.

2. For any vertex $x \notin T$ which is labeled and unscanned:
   - (a) Label every vertex y as $(x,-)$ if $(y,x) \in \Gamma$, y is - unlabeled, $f(y,x) = 1$, $y \notin S$, and
     - (i) x is + labeled, + unscanned, and - unlabeled, or
     - (ii) x is - labeled and - unscanned.

     If in (ii) x is + labeled and + unscanned, then add "!" to the label on y. If y is + labeled and + unscanned, then erase the + label.
   - (b) Label every vertex y as $(x,+)$ if $(x,y) \in \Gamma$, y is + unlabeled, or y is + labeled and in T, $f(x,y) = 0$, $y \notin S$, and
     - (i) $x \in S$, x is + labeled and + unscanned, or
     - (ii) $x \notin S$, x is + labeled, + unscanned, - unlabeled and $f(x,V) = 0$, or
     - (iii) $x \notin S$, x is - labeled and - unscanned.

     If in (iii) x is + labeled and - unscanned, then add "!" to the label on y.
   - (c) In cases a(i), b(i) and (ii), + scan x. In cases a(ii) and b(iii), - scan x.
3. Repeat step 2 until either no more labels can be assigned and no $t \in T$ is labeled or some $t \in T$ is labeled.
   - (i) In the former case, terminate the procedure.
   - (ii) In the latter case, use Routine B to obtain a feasible flow path, if possible. Then, if any further labeling is possible, go to step 2; otherwise, go to step 1.

Routine B - Flow Change

1. Let $z = t$ and go to step 2.
2. (a) If the label on z is $(w,+)$ and
     - (i) $f(w,z) = 0$, increase $f(w,z)$ by 1 and go to step 3; or
     - (ii) $f(w,z) = 1$, go to step 5.

   (b) If the label on z is $(w,-)$ and
     - (i) $f(z,w) = 1$, decrease $f(z,w)$ by 1 and go to step 3; or
     - (ii) $f(z,w) = 0$, go to step 5.
3. If $q \in S$, then return to step 3(ii) of Routine A. Otherwise, go to step 4.
4. If in step 2 the label $(w,+)$ or $(w,-)$ is followed by an "!", then replace the current value of z by w and go to step 2(b). If in step 2 the label is not followed by an "!", then replace z by w and go to step 2.
5. No feasible flow path is possible. Restore the old flow pattern existing before the application of Routine B and go to step 3(ii) of Routine A.

B. An Algorithm Containing GFLA which Sequentially Yields Subclusters From a Parent Cluster with the Chaining Effect

Step 1: Specify a splitting criterion; namely, specify the reference size of a desirable articulation set, k, and the reference size of a desirable subcluster, m, where k and m are two positive integers.

Step 2: From the graph consisting of the set of vertices under investigation, choose two small connected subgraphs which are about farthest from each other and which have more than k adjacent vertices each. Consider the two sets of vertices in these two subgraphs as the multiple sources, S, and the multiple sinks, T.

Step 3: Apply the generalized Frisch's labeling algorithm to the set of vertices under investigation. Obtain the sets of labeled and unlabeled vertices and the articulation set, denoted as L, U and A. Let $|L|$, $|U|$ and $|A|$ be their sizes. List L, U, A, $|L|$, $|U|$ and $|A|$.

Step 4:

(a) If $m \leq |L| < 2m$, go to step 5.

(b) If $|L| \geq 2m$ and
  - (i) if $|U| \geq 2m$, store L, and L needs further investigation. Go to step 5.
  - (ii) if $|U| < m$, go to step 6.

(c) If $|L| < m$ and
  - (i) if $|L| \approx |S| + k$, go to step 6.
  - (ii) if $|S| + k \ll |L| < m$, go to step 5.

Step 5: Break the set of vertices under investigation into two sets L and U.

(a) If $m \leq |U| < 2m$, go to step 7.

(b) If $|U| \geq 2m$, store U, and U needs further investigation. Go to step 7.

(c) If $|U| < m$, go to step 7.

Step 6: Remove all vertices in the flow paths and find any connected subgraphs which have each at least m vertices. These are the candidate subclusters. For each candidate subcluster find the adjacent vertices in the flow paths. If the number of adjacent vertices is k or less, then list the subgraph and the adjacent vertices; otherwise store the subgraph and the adjacent vertices for further investigation. Remove all candidate subclusters, if any, from the original set of vertices under investigation. If the number of remaining vertices is less than 2m, list these vertices and go to step 7; otherwise apply the generalized Frisch's labeling algorithm to these remaining vertices by choosing the two sets of vertices in each pair of flow paths (excluding the original sources and sinks) as the new sources and the new sinks, respectively. Combine the two flow paths and other labeled vertices if the number of new flow paths is either more than k or about equal to the size of the smaller flow path. In this way, obtain all maximal subgraphs. List these maximal subgraphs and any associated articulation sets. Then go to step 7.

Step 7: Check whether there are any candidate subclusters or any subgraphs which need further investigation. If no, go to step 8; if yes, then

(i) for each candidate subcluster, treat its associated set of adjacent vertices in the flow paths as multiple sinks and find, as multiple sources, a small connected set of vertices which is about farthest from the sinks and which has more than k adjacent vertices. Then go to step 3.

(ii) for each connected subgraph of size greater than or equal to 2m, go to step 2.

Step 8: For each articulation set, find the subgraphs which surround the vertices of it. Then determine which subgraphs are legitimate subclusters.

Step 9: Combine each subgraph of size less than m with one of its neighboring subgraphs in such a way to produce any new possible subclusters.

Step 10: Stop.

REFERENCES

[1] J. G. Auguston and J. Minker, "An Analysis of Some Graph Theoretical Cluster Techniques," J. ACM, Vol. 17, No. 4, pp. 571-588, Oct. 1970.

[2] R. E. Bonner, "On Some Clustering Techniques," IBM J., pp. 22-32, Jan. 1964.

[3] M. Wirth, G. F. Estabrook, and D. J. Rogers, "A Graph Theory Model for Systematic Biology, With an Example for the Oncidiinae (Orchidaceae)," Systematic Zool., Vol. 15, pp. 59-69, 1966.

[4] G. F. Estabrook, "A Mathematical Model in Graph Theory for Biological Classification," J. Theor. Biology, Vol. 12, pp. 297-310, 1966.

[5] R. F. Ling, "Cluster Analysis," Tech. Rept., Yale U., Conn., 1971.

[6] N. Jardine and R. Sibson, "The Construction of Hierarchic and Non-Hierarchic Classifications," Computer J., Vol. 11, pp. 177-184, 1968.

[7] C. J. Jardine, N. Jardine, and R. Sibson, "The Structure and Construction of Taxonomic Hierarchies," Math. Biosci., Vol. 1, pp. 173-179,1967.

[8] N. Jardine and R. Sibson, "A Model for Taxonomy," Math. Biosci., Vol. 2, pp. 465-482, 1968.

[9] M. J. Sheperd and A. J. Willmott, "Cluster Analysis on the Atlas Computer," Computer J., Vol. 11, pp. 57-62, 1968.

[10] C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. Comp., Vol. C-20, pp. 68-86, Jan. 1971.

[11] F. Harary, Graph Theory, Addison-Wesley, Mass., 1965.

[12] C. Berge, Theory of Graph and Its Applications, John Wiley, N.Y., 1962.

[13] L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton U. Press, N.J., 1962.

[14] O. Ore, Theory of Graphs, Am. Math. Soc. Colloq. Publ., Providence, R.I., 1962.

[15] L. R. Ford and D. R. Fulkerson, "Maximal Flow Through a Network," Canad. J. Math., 8, pp. 399-404, 1956.

[16] J. T. Frisch, "An Algorithm for Vertex-Pair Connectivity," Intl. J. Control, Vol. 6, No. 6, pp. 579-593, 1967.

[17] Z. Chen and K. S. Fu, "On the Connectivity of Clusters," (to be publ.).

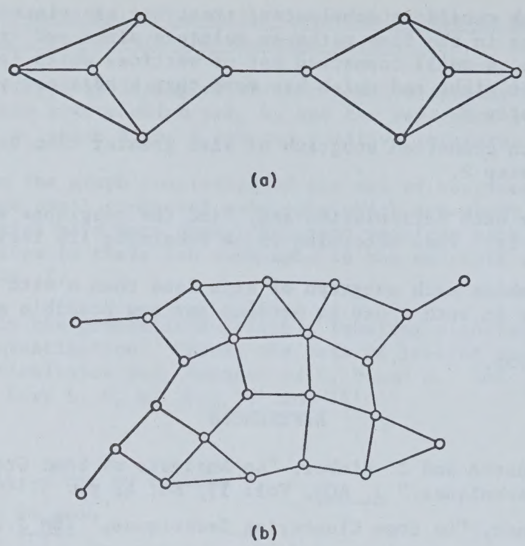


FIGURE 1. (a) The 1-connectedness of a cluster reveals the chaining effect.
(b) The connectivity of a cluster is ruined by a few vertices of degree 1.

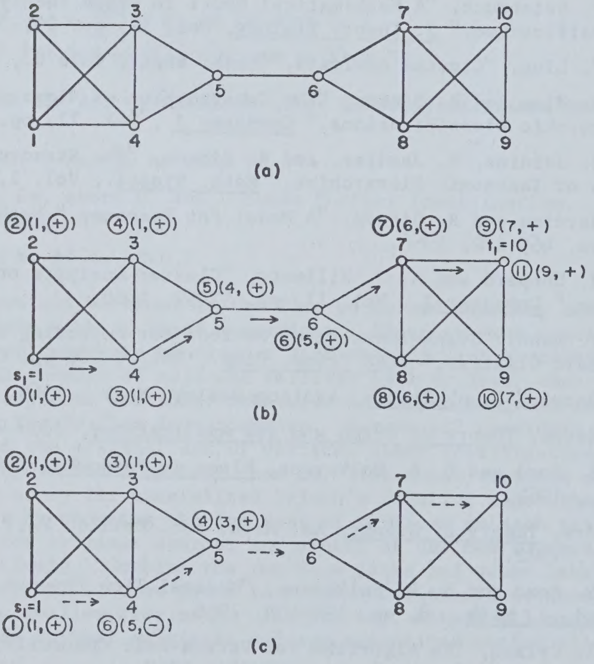

2
3
7
10
5
6
1
4
8
9
(a)
②(1,⊕)
④(1,⊕)
⑦(6,⊕)
⑨(7, +)
2
3
7
t_1=10
⑪(9, +)
⑤(4, ⊕)
5
6
⑥(5,⊕)
s_1=1
4
8
9
①(1,⊕)
③(1,⊕)
⑧(6,⊕)
⑩(7,⊕)
(b)
②(1,⊕)
③(1,⊕)
2
3
7
10
④(3,⊕)
5
6
s_1=1
4
8
9
①(1,⊕)
⑥(5,⊖)
(c)

FIGURE 2. (a) The graph for the given example in the text.
(b) The construction of a flow path in the graph.
(c) The final result of the labelling process.

DESIGN OF THRESHOLD ELEMENTS USING A POLYNOMIAL-THRESHOLD APPROACH*

M. H. LOEW† and K. S. FU
School of Electrical Engineering, Purdue University, Lafayette, Indiana 47907

ABSTRACT

A method is presented which reduces the number of thresholds required by a multi-threshold threshold element to realize a function of a nonbinary variable, as would arise in certain nonparametric pattern-recognition techniques.

Polynomials (often of only first- or second-degree) replace selected thresholds, and predetermined bounds enable the designer to determine in advance the value of the trade. Illustrative examples are presented and discussed.

1. INTRODUCTION

The basic selection unit (BSU) proposed by Henrichon and Fu [1] for nonparametric pattern classification contains, as its final element, a multi-threshold threshold unit. The threshold unit compares its input with the set of thresholds computed by the empirical processing algorithm (EPA), and either assigns one of the classes $\omega_1,\ldots,\omega_n$, or activates a selection line. The selection line corresponds to class ω_0 (indecision), and thus can be treated as an (n+1)st class in the following discussion. This approach has been successfully applied to problems of crop classification and medical diagnosis [1,7]. This paper considers a more flexible approach to the realization of the BSU's final element; a combination of threshold elements and polynomial-generating units is shown to perform the same function. A systematic but flexible procedure is outlined which would enable the system designer to choose between several combinations of elements.

2. THE POLYNOMIAL-THRESHOLD APPROACH

The empirical processing algorithm yields a string of class labels for each feature and a set of thresholds to which the value of an incoming feature is compared. This string can be represented by an axis of increasing magnitude of the feature (Fig. 1). In general, repetitions of classes will occur along the axis, with the EPA insuring that no two adjacent entries are the same.

Consider a mapping from the feature axis onto a line (the y-axis) which is partitioned into as many regions as there are classes. These n+1 regions are separated by n thresholds whose values depend on the procedure given below. The mapping is specified by one or more polynomials each of whose domains spans the portion of the feature axis specified by the EPA. These polynomials map every occurrence of each class into just one region per class along the y-axis. Both the ordering of the classes along the y-axis and the values of the thresholds $T_1',T_2',\ldots,T_n'$ are arbitrary, although a method for choosing them is suggested below. An example will help illustrate the procedure (Fig. 2).

Step 1. Choose an ordering for the classes along the y-axis. For example, the natural ordering $\omega_0,\omega_1,\omega_2,\omega_3$ proceeding downward along the axis, is used here.

Step 2. Assign thresholds $T_1',T_2',\ldots,T_n'$ at arbitrary points along the axis to separate the regions.

Step 3. Construct a grid by drawing horizontal lines through $\{T_i'\}_{i=1}^{n}$ and vertical lines through each of the EPA thresholds and shade the rectangles lying at the intersections of each of the EPA regions with its

---

*This work was supported by AFOSR Grant 69-1776.

†Now with Analytic Services Inc. (ANSER), Falls Church, Virginia 22041.

corresponding y-axis region.

Step 4. Determine* a polynomial which will fit each of the continuous groups of shaded rectangles and pass through the corners separating them. A continuous group of rectangles consists of the largest set of rectangles such that every consecutive pair shares a common point (at corners). Thus, in Fig. 2 the continuous groups are $\omega_1\omega_2$ (separated by T_1) and $\omega_1\omega_2\omega_3$ (separated by T_5 and T_6), and the polynomials are both first-degree.

Step 5. Identify thresholds T_i at which discontinuities occur; that is, where a continuous group or an isolated rectangle begins and ends. In this example the discontinuities, called jumps, occur at T_2, T_3, T_4, and T_7.

Step 6. Now construct a modified BSU consisting of an initial threshold unit (whose thresholds are exactly those at which the jumps occur), polynomial-generation units, and a final threshold unit. When a feature value falls in the domain of one of the polynomials, the initial threshold unit activates the appropriate polynomial unit which computes the value of y and uses it as the input to the final threshold unit. If a feature value falls elsewhere, it is classified immediately by the initial threshold unit. The modified BSU structure for this example is shown in Fig. 3, where circled entries indicate final decisions.

This new structure requires six thresholds and two first-degree polynomials, as compared to the seven thresholds which would have been required had the original BSU structure been used. But if we now impose some constraints on both the ordering of classes along the y-axis and the values of the T_i', we will be able to effect greater savings in the number of required processing units. The next section indicates how these constraints may be chosen.

3. THE ADJACENCY MATRIX

One approach to the reduction of complexity of the modified BSU is to reduce the number of jumps—and thus the number of thresholds in the initial unit—by ordering the classes along the y-axis.

The adjacency matrix is a useful way to represent the relationships between entries on the feature axis (Fig. 1). Given a string of p labels (generally with class repetitions) and the associated p-1 thresholds resulting from the EPA, and assuming the n+1 classes are $\omega_0, \omega_1, \ldots, \omega_n$, we construct the matrix A as follows.

1. Label the n columns as $\omega_0, \omega_1, \ldots, \omega_{n-1}$ and the n rows as $\omega_1, \omega_2, \ldots, \omega_n$ (see Fig. 4).

2. In each location a_{ii} $(i=1,2,\ldots,n-1)$ place an X.

3. Start at the left end of the string and, for each element ω_i and its successor ω_j $(i \neq j)$, enter a count either in cell a_{ij} if $i>j$ or in cell a_{ji} if $j>i$ $(i=1,\ldots,n;\ j=0,\ldots,n-1)$. Enter zeros in all unfilled cells below the diagonal. This matrix, which is strictly below the diagonal of X's, is the adjacency matrix. Note that the sum of the a_{ij} should be p-1, the total number of adjacencies in the string, since no adjacent elements are of the same class.

Inspection of Fig. 4 indicates that class ω_1 is adjacent twice to both classes ω_2 and ω_3, never to ω_0, and that all other adjacencies occur once. Thus, in the class ordering along the y-axis it would be advantageous to position ω_1 between ω_2 and ω_3 to make each of those four transitions smooth. At this point, then, the ordering is $\omega_2\omega_1\omega_3$, and only ω_0 must be placed. The adjacency matrix indicates that ω_0 is adjacent once to ω_2 and once to ω_3, so one jump will occur regardless of the position of ω_0. Arbitrarily position it next to ω_2 and construct a new set of axes similar to that in Fig. 2 (see Fig. 5). The original set of EPA thresholds $T_1, T_2, \ldots, T_{p-1}$ is unchanged, but the set of T_i' remains arbitrary. Fig. 5 indicates that there are three continuous groups, implying two jumps and three polynomials. Here again, each polynomial is of the first degree, and our final structure requires a total of five thresholds and three first-degree polynomials, as compared to six and two, respectively, for the structure of Fig. 2; a threshold has been saved at the expense of adding a polynomial. The next section presents a method which allows the designer to vary the numbers of polynomials and thresholds required, subject to certain lower bounds.

---

*Section 4 describes some procedures for determining the various polynomials.

4. APPLICATION OF THE ADJACENCY MATRIX AND CONSTRUCTION OF POLYNOMIALS

4.1 Use and Properties of the Adjacency Matrix

For each jump which appears on the display for the final y-axis ordering, a threshold will be required in the initial unit. It is worthwhile, therefore, to examine the properties of the adjacency matrix with the intent of minimizing the number of jumps. Continuing the notation used earlier, we have n+1 for the number of classes, and p for the total number of entries along the x-axis of the display.

A circled cell of the adjacency matrix will mean that the class labels corresponding to that cell will be placed adjacent to each other on the y-axis. Adopting this procedure, we can observe the following properties.

Property 1. There will be at most n circled cells, since all n+1 classes must appear on the y-axis, and thus there will be n adjacencies. (The fact that n is the maximum, rather than the exact, number of circled cells is shown in Sec. 5.2.)

Property 2. The number of jumps equals the sum of the uncircled counts, and, thus, the entries should be circled which will maximize the total circled count.

Property 3. No more than two cells having a particular label as one of their coordinates can be circled, because a given class label can be adjacent to at most two other labels on the y-axis.

Property 4. Lemma: When $n \geq 3$ and no zeros appear in the adjacency matrix, the final display will have at least n+1 jumps.

Proof: No zeros implies that all possible pairs of classes exist on the x-axis. Hence, regardless of the final y-axis ordering, there will be, for at least one occurrence of each symbol, another symbol adjacent on the x-axis which is not adjacent on the y-axis. [For n=2, no zeros implies at least one jump; for n=1 (the two-class case), no jumps will appear.] Q.E.D.

Example 4.1. The string of class labels shown in Fig. 6(a) can be assumed to be the result for feature F_i of the application of the EPA to a five-class problem and would require a 21-threshold BSU for implementation. From the procedure discussed previously, the adjacency matrix of Fig. 6(b) is constructed. Then, n=4 cells are circled which will give the greatest total of included adjacencies. The ω_3-ω_1 cell, with four entries, is chosen first, and then we note that four cells remain with three adjacencies each. Only three of the cells can be chosen, however, and in this case, any three of the four are acceptable; i.e., no choice would violate Property 3 above. We choose here the cells ω_0-ω_2, ω_0-ω_4, and ω_2-ω_3. The grid which will determine the mapping can now be constructed [Fig. 6(c)]. The circled cells are used to establish the ordering along the y-axis by noting that ω_3 must be adjacent to both ω_1 and ω_2, implying the sequence $\omega_1\omega_3\omega_2$; and that ω_0 must be adjacent to both ω_2 and ω_4, implying the sequence $\omega_2\omega_0\omega_4$. The two sequences taken together, then, yield $\omega_1\omega_3\omega_2\omega_0\omega_4$ (or its reversal, $\omega_4\omega_0\omega_2\omega_3\omega_1$) as the y-axis ordering.

The eight jumps indicated in Fig. 6(c) at T_5, T_6, T_7, T_{11}, T_{18}, T_{19}, T_{20}, and T_{21} correspond as predicted to the uncircled entries in the adjacency matrix. The grid now contains three continuous groups: between the origin and T_5, between T_7 and T_{11}, and between T_{11} and T_{18}. The polynomials needed to map them onto the y-axis and the associated determination of the $\{T_i'\}$ are considered next.

4.2. The Mapping Polynomials.

The first approach might be to construct a polynomial whose curve passes through each of the "inner" corners of a continuous group and which has inflection points properly located within the individual blocks of the continuous group. In the first continuous group of Fig. 6(c), for example, the curve must pass through the points (T_1,T'_4), (T_2,T'_4), (T_3,T'_4), and (T_4, T'_4); it must also have relative minima within the two lower blocks and a relative maximum within the center upper block. Those conditions alone imply a polynomial whose degree is at least four. While it is true that no values have yet been assigned to T'_3 and T'_4, that an upper limit for the y-axis has not been chosen, and that this freedom may simplify the polynomial, these decisions will not reduce the minimum degree of the polynomial. Determination of the coefficients of the polynomials will not be considered here because applicable techniques are well-known (e.g., [2], [3]).

An alternative approach, which reduces the degrees of the required polynomials, is that of introducing "artificial" jumps. By breaking up continuous groups with selected new thresholds, the method reduces the degrees of the mapping polynomials. For example, in the first continuous group of Fig. 6(c), a threshold could be placed at either T_2 or T_3 with the result that the mapping of the five blocks would now require one polynomial of first- and one of second-degree. (A parabola mapping the three blocks from the origin through T_3, and a straight line mapping the two between T_3 and T_5; or the parabola mapping from T_2 to T_5, and the line mapping from the origin through T_2.)

Techniques for the construction of parabolas and straight lines using the flexibility of the $\{T'_i\}$ are not problem-dependent; we present briefly some of the results found in [4].

A parabola whose axis is parallel to the y-axis can always be constructed in the situation where three consecutive blocks of a continuous group are arranged so that the two end blocks have the same class label; for example, origin - T_3, T_2 - T_5, T_7 - T_{10}, T_8 - T_{11}, etc., in Fig. 6(c). The construction of a parabola imposes a weak constraint on only one of the $\{T'_i\}$, and this cannot be considered a major restriction.

Any two adjacent blocks in a continuous group can be mapped by a straight line, regardless of the $\{T'_i\}$. Any number of monotonically increasing or decreasing (in y) consecutive blocks in a continuous group can be mapped by a straight line if restrictions may be placed on the subset of the $\{T'_i\}$ involved.

Application of the procedures outlined above is summarized with respect to Fig. 6(c) by Table 1, which lists the numbers of the different kinds of elements that would be required for various implementations of the EPA results. The lower bound of 12 thresholds for the modified structures is due to the eight jumps (which always are required), and to the four T'_i.

It is apparent from Table 1 that a tradeoff occurs when going from the straight BSU structure to a modification along the lines of Fig. 3. The required number of thresholds is reduced at a cost of adding some polynomial-generation units. The value of the trade, of course, depends on the costs

TABLE 1. POSSIBLE STRUCTURES FOR A GIVEN EPA OUTPUT

Structure	Thresholds	Polynomials		
		1st Degree	2nd Degree	Degree > 2 (Min. degree in parentheses)
Straight BSU	21			
Modification 1	17	7		
Modification 2	16	1	4	
Modification 3	12			3: 1(≥ 4),1(≥ 3),1(≥ 5)

of the units involved, physical considerations, and many other factors which designers routinely evaluate.

Some steps toward a more formal setting for analysis of the value of the adjacency matrix are presented in the next section.

5. SOME CONDITIONS ON THE UTILITY OF THE METHOD

5.1. The Goal

The objective of the adjacency-matrix construction and manipulation, as defined previously, is to reduce the number of jumps, or discontinuities, which occur as a result of the mapping procedure. We will refer collectively to the steps which follow the actual construction of the adjacency matrix as the *reduction* process. This was motivated by the fact that each jump in the final array leads to the need for an additional threshold in the initial threshold unit; and although the number of thresholds required by the jumps is smaller than the number which would have been needed in the absence of this procedure, it must be remembered that n additional thresholds are required by the procedure for the final threshold unit (see Fig. 3), since there are $n+1$ classes.

In other words, the number of jumps plus n must be less than the original number of adjacencies, $p-1$, if the total number of thresholds is to be reduced. Letting j equal the number of jumps, the condition is $j+n < p-1$. Rewriting this as $n < (p-1)-j$, we can recognize $(p-1)-j$ as the total *circled* count in the adjacency matrix; that is, the number of jumps which was eliminated by the arrangement of classes along the y-axis. The condition now becomes

$$(\text{number of circled counts}) \geq n + 1 \quad . \tag{1}$$

Satisfaction of (1) is therefore a major goal of the procedure. To discuss the conditions under which (1) holds, we define certain other characteristics of the adjacency matrix in the next section.

5.2. Structure of the Adjacency Matrix

A *zero cell* is a cell in the matrix which has no counts. That is, the class labels of the cell's row and column are never adjacent on the x-axis. We will call the matrix *uniform* if all nonzero cells have the same number of counts.

A set of cells in the adjacency matrix will be called *compatible* if the arrangement of classes along the y-axis that is implied by the circling of all the cells in that set is possible. Property 3 in Sec. 4.1 is one example of a necessary condition for compatibility. Other conditions applicable to compatibility are discussed in Sec. 5.4. Note, however, that a particular cell's being a member of a compatible set is dependent strictly on its location and not on its contained count.

Property 1 in Sec. 4.1 can now be reconsidered. It states that there can be *at most* n circled cells; that is, that the largest compatible set for any adjacency matrix must have n or fewer elements. In the absence of constraints on which elements can be circled, the upper bound of that statement follows, since circling a cell is equivalent to pairing its two class labels along the y-axis; and we cannot specify more than n adjacencies on a linear (i.e., noncyclic) group of $n+1$ items. The inequality in Property 1 can be demonstrated by use of Property 3, for example. Suppose the only nonzero cells of an adjacency matrix are $\omega_1\omega_0, \omega_2\omega_0, \ldots, \omega_n\omega_0$. No set of three or more of these cells is compatible because circling any three or more of them would impose impossible conditions on the y-axis ordering—for example, ω_0 could not be simultaneously adjacent to ω_1, ω_2, *and* ω_4 along the y-axis. Thus, of the n nonzero cells in the example, only two (in this case any two) can be circled; this leads to the inequality of Property 1.

Since each of the $n+1$ classes must appear at least once on the x-axis, there must be at least n distinct adjacencies, implying that

$$\text{(number of nonzero cells)} \geq n \quad , \tag{2}$$

which in turn implies

$$\text{(number of zero cells)} \leq \tfrac{1}{2}n(n - 1) \quad , \tag{3}$$

since the total number of cells is $\frac{1}{2}n(n+1)$.

We now establish a condition which holds regardless of compatibility considerations.

Theorem 5.2. Of the set of all adjacency matrices for n+1 classes which have p-1 total counts (adjacencies), none will have more jumps remaining after reduction than that matrix which is uniform in only the circled cells. (The proof of this theorem can be found in [4].)

It will be shown in Sec. 5.4 that if no zero cells appear in the matrix, then it will always be possible to find a compatible set of size n. It is also shown in [4] that in the no-zeros, nonuniform case, the reduction process will always be worthwhile.

5.3. Results for Some Special Cases

It is not yet known, in general, how to determine in advance (that is, with knowledge only of p-1, n, number of zero cells, and uniformity) whether the reduction process will yield an advantageous structure. The result at the end of the previous section provides some information under very specific conditions. This section deals with some additional special cases and determines bounds which indicate when the procedure should be used.

5.3.1. Uniform Case—Realizability. We begin with two assumptions: the adjacency matrix is uniform, and a compatible set of n cells exists; that is, n cells can be circled. Thus, the matrix will be uniform in only the circled cells, satisfying the conditions of Theorem 5.2, and therefore will have more jumps than any rearranged version of it would have, making it the worst-case arrangement. No matrix in which n cells can be circled will have more jumps after reduction than this one. We would like to find the conditions for this matrix under which the reduction process will yield fewer thresholds than the original structure.

Let there be a matrix having p-1 adjacencies for n+1 classes, with m zero cells [$0 \leq m \leq \frac{1}{2}n(n-1)$, from (3)], n circled cells, and $\frac{1}{2}n(n+1)$ total cells. The matrix is uniform, so

$$\text{(number of counts/cell)} = \frac{p - 1}{\frac{1}{2}n(n + 1) - m}$$

and

$$\begin{aligned}\text{(number of jumps)} &= \text{(total)} - \text{(circled)} \\ &= p - 1 - n\left(\frac{p - 1}{\frac{1}{2}n(n + 1) - m}\right) \quad .\end{aligned}$$

Now, the total number of thresholds, k', is the number of jumps plus n (the number of thresholds required along the y-axis), or

$$k' = p - 1 - n\left(\frac{p - 1}{\frac{1}{2}n(n + 1) - m}\right) + n \quad .$$

The condition under which $k' < p - 1$ can now be written as

$$n + 1 \leq \frac{-1 + \sqrt{8(m+p)-7}}{2} \quad . \tag{4}$$

Equation (4) is plotted in Fig. 7 for m=0 and several values of n+1 and p. Any point lying on or below the "realizability" curve indicates that the adjacency matrix it represents should be reduced. The curve moves to the left by one unit of p for each zero cell in the matrix, as (4) indicates.

5.3.2. Uniform Case—Straight Lines. The basis of the reduction procedure described in Sec. 4 was that, to reduce the required number of thresholds, polynomials can be constructed to span two or more consecutive class occurrences, existing between jumps, along the x-axis. The polynomial for a straight line, $y=mx+b$, can, with proper normalization of the coefficients, be physically realized very cheaply. This fact motivates the present section, in which we treat the question of determining in advance whether it will be worthwhile to introduce sufficient additional jumps to insure that no polynomial of degree higher than one will be needed.

We again approach the problem from the worst-case point of view by asking: (1) What is the maximum number of new jumps that would be required to guarantee straight-line realizations? and (2) Under what conditions will the addition of these jumps still result in a reduction of the number of thresholds required?

In the following, when a consecutive set of n class labels appears between two jumps, we refer to the set as "a group of n occurrences."

Consider a group of three occurrences. If we wish to add additional jumps so that only first-degree polynomials are required, we would add a single jump between the first and second or between the second and third class labels, thus partitioning the group of three into one group of two and one of one. We have shown previously that any group of two occurrences can be spanned by a straight line. The group of one would be classified immediately by the initial threshold unit. For the original group of three then, just one additional jump is required.

Now consider a group of four occurrences. A single additional jump, placed between the second and third class labels, partitions the group into two groups of two, each of which can be realized by a straight line. We conclude that for either three or four occurrences, a single new jump is sufficient. Continuing by induction, we are led immediately to the following.

Lemma: With respect to partitioning a group of occurrences into groups realizable by straight lines, the number of new jumps required for an odd number of occurrences, q, is the same as that required for $q+1$ occurrences; namely, $(q-1)/2$.

Thus, we can maximize the number of additional jumps (as required by the worst-case analysis) by maximizing the number of groups which have odd numbers of occurrences. A construction will clarify this concept. Let there be a string of p class occurrences on the x-axis, and let there be r original jumps, which partition the x-axis into r+1 groups. Then a situation requiring the largest number of additional jumps would be the one in which there are r groups of one occurrence each (requiring no new jumps) and one group of p-r occurrences (requiring $[(p-r-1)/2]$ new jumps*).

That this yields the maximum may be seen by considering a redistribution of the group of p-r occurrences. If one occurrence were shifted from this group to one of the r single-occurrence groups, no new jump would be required at the group that now has two occurrences, but if p-r were odd, losing one occurrence would reduce by one the number of jumps required within it. A second shift would not change the number of jumps required, but a third would, and so on. Thus, any redistribution would serve only to hold constant or to reduce the number of additional jumps required.

We are now ready to determine an answer to the second question posed at the beginning of this section. As in the previous section, define the adjacency matrix as representing an array of p-1 adjacencies (p occurrences) and n+1 classes, and as having m zero cells $[0 \le m \le \frac{1}{2}n(n-1)]$, and $\frac{1}{2}n(n+1)$ total cells. Assume that a compatible set of size n exists, and that the matrix is uniform. Then,

$$(\text{number of counts/cell}) = \frac{p - 1}{\frac{1}{2}n(n + 1) - m} ,$$

and

---

*We use $[x]$ to indicate the largest integer $\le x$.

$$(\text{number of jumps}) = p - 1 - n\left(\frac{p - 1}{\frac{1}{2}n(n + 1) - m}\right)$$

$$= (p - 1)\left(\frac{\frac{1}{2}n(n - 1) - m}{\frac{1}{2}n(n + 1) - m}\right) .$$

Then, since the total number of groups is equal to one more than the number of jumps, we can split the set of groups into

$$(p - 1)\left(\frac{\frac{1}{2}n(n - 1) - m}{\frac{1}{2}n(n + 1) - m}\right) \text{ groups of one occurrence each}$$

and one group of

$$p - 1 - (p - 1) \left(\frac{\frac{1}{2}n(n - 1) - m}{\frac{1}{2}n(n + 1) - m}\right) = \frac{n(p - 1)}{\frac{1}{2}n(n + 1) - m} \quad \text{occurrences.}$$

This is exactly the construction described in the worst-case situation above. The total number of thresholds, k', is now the sum of the original number of jumps, n (from the final unit), and half the number of occurrences in the last group of the construction.

$$k' = (p - 1)\left(\frac{\frac{1}{2}n(n - 1) - m}{\frac{1}{2}n(n + 1) - m}\right) + n + \frac{1}{2}\left(\frac{n(p - 1)}{\frac{1}{2}n(n + 1) - m}\right)$$

$$= (p - 1)\left(\frac{n^2 - 2m}{n(n + 1) - 2m}\right) + n \quad .$$

Solving the inequality $k' < p - 1$, we obtain the condition

$$n + 1 \le \frac{- 1 + \sqrt{4(2m+p) - 3}}{2} \quad . \tag{5}$$

Equation (5) is also plotted in Fig. 7 for m=0 and various values of n+1 and p. Any point lying on or below the "straight lines" curve indicates that a classifier realization using only threshold units and first-degree polynomials would require fewer thresholds than the straight-threshold structure. For each zero cell in the adjacency matrix, the curve moves to the left by two units of p. Similarly, if the designer requires a larger difference between k' and p-1, he may increase it by one for each shift by one to the right of either of the curves in Fig. 7.

5.4. A Graph-Theoretic Interpretation

If we reexamine the adjacency matrix in the light of some results from graph theory, we find that certain properties of the structure of the matrix are more readily discerned. The correspondence is apparent if we let the vertices of the graph represent the class labels of the adjacency matrix and let the edges incident to pairs of vertices represent the adjacencies within the matrix. Each is then completely specified by the other.

Recalling that the adjacency matrix was generated from the original string of class labels along the x-axis, we note that treating multiple adjacencies between pairs of classes as distinct directed edges makes the string an Euler path*[5] or a cycle, depending on whether the first class label on the axis is different from, or the same as, respectively, the last label on the axis.

Since no repetitions of class labels are allowed along the y-axis, the ordering along the axis will be represented by a spanning tree in the graph. Thus, a one-to-one correspondence exists between the circled cells of the adjacency matrix and the branches of the spanning tree, in that each indicates an adjacency on the y-axis. We can use this relationship to explore the conditions on compatibility: What determines the number of cells which can be circled in the adjacency matrix? First, the fact that a tree which spans a graph of n+1 vertices has exactly n branches means that it will never be possible to circle more than n cells in the matrix. Second, a tree may not contain a circuit, implying that the circled cells in the matrix may not be cyclic; for example, the adjacencies $\omega_0\omega_1$, $\omega_1\omega_2$, $\omega_2\omega_0$ could not all be circled.

---

*An Euler path or cycle is a path or cycle that traverses each edge in a graph exactly once.

Thus, the existence of a tree in the graph associated with an adjacency matrix is a necessary condition for the existence of a compatibility set of size n. But it is not a sufficient condition for the following reason. In addition to forbidding the cyclic arrangement of classes on the y-axis, the reduction process also prevents a class label from being adjacent to more than two other labels. This is equivalent to restricting the number of edges incident to a vertex in the tree to two or fewer. This kind of restricted path is called an Eulerian chain by Berge [6]. An example of a set which is incompatible because it violates this condition is $\omega_0\omega_1$, $\omega_0\omega_2$, $\omega_0\omega_3$; we cannot circle all three of them.

New implications now arise from previously noted conditions. An adjacency matrix which has the maximum number of zero cells $[\frac{1}{2}n(n-1)]$ has a tree as its graph. So we need only check the matrix to determine if its graph (tree) is an eulerian chain. This is simply done by inspecting the rows and columns of the matrix for more than two nonzero entries in any one row or column.

A Hamiltonian path or circuit is a path or circuit that passes through each of the vertices of a graph exactly once [5]. If a graph has a Hamiltonian path, we know that a compatible set of size n exists, since that path will include n adjacencies and can be translated to the y-axis. A sufficient condition for the existence of a Hamiltonian path in a graph is that the graph be complete; that is, that every pair of vertices be connected by at least one edge [5]. But that condition is exactly met by an adjacency matrix with no zero cells, enabling us to say that such a matrix always has a compatibility set of size n, proving a statement made in Sec. 5.2.

We have shown in this section that certain results from the theory of graphs have direct application to the reduction techniques described here. There seems to be considerable opportunity for future development in this area.

6. IMPLICATIONS FOR THE DESIGNER

The procedures presented here are intended to be guides for considering tradeoffs in the design of classifiers along the lines of the BSU; no optimality is claimed for them. Only the designer, aware of the space, cost, reliability, and other factors pertinent to his application, can decide whether the use of a first-degree-polynomial generator is preferable to that of one or more threshold units. As more knowledge is obtained about the adjacency matrix's structure, more general constraints on, and other realizations of, a BSU-type classifier should become available.

REFERENCES

[1] E. G. Henrichon, Jr. and K. S. Fu, "A Nonparametric Partitioning Procedure for Pattern Classification," IEEE Trans. Computers, vol. C-18, No. 7, pp. 614-624, July 1969.

[2] R. W. Hamming, Numerical Methods for Scientists and Engineers. New York: McGraw-Hill, 1962, Chap. 8.

[3] I. S. Sokolnikoff and R. M. Redheffer, Mathematics of Physics and Modern Engineering. New York: McGraw-Hill, 1958, Chap. 9.

[4] M. H. Loew and K. S. Fu, Computer-Aided Medical Diagnosis Using Sequential Pattern-Recognition Techniques, Purdue University, School of Electrical Engineering, Tech. Rept. TR-EE-72-14, May 1972.

[5] C. L. Liu, Introduction to Combinational Mathematics. New York: McGraw-Hill, 1968.

[6] C. Berge, The Theory of Graphs and Its Applications. New York: Wiley, 1962.

[7] K. S. Fu and M. H. Loew, "Automatic Medical Diagnosis Using Nonparametric Sequential Classification Procedures," AGARD Conference Proceedings No. 94 on Artificial Intelligence. Neuilly-sur-Seine, France: NATO, 1971, pp. 10-1—10-9.

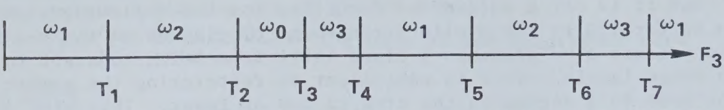

ω_1
ω_2
ω_0
ω_3
ω_1
ω_2
ω_3
ω_1
F_3
T_1
T_2
T_3
T_4
T_5
T_6
T_7

FIG. 1. AXIS OF VALUES FOR FEATURE F_3, SHOWING FOUR CLASSES AND SEVEN THRESHOLDS

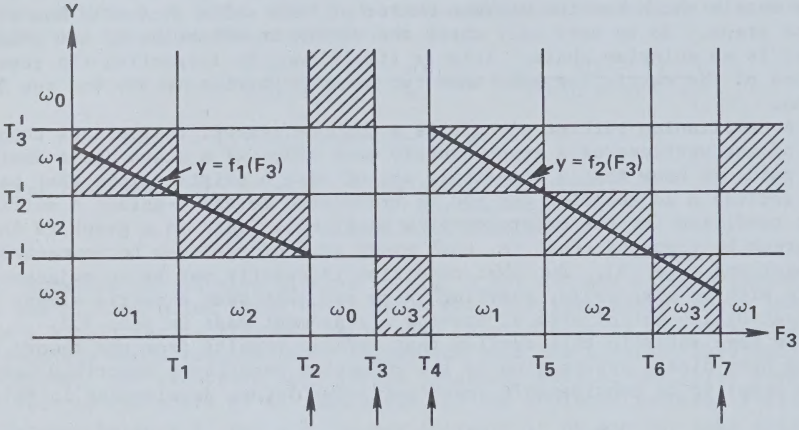


FIG. 2. EXAMPLES OF CONTINUOUS GROUPS

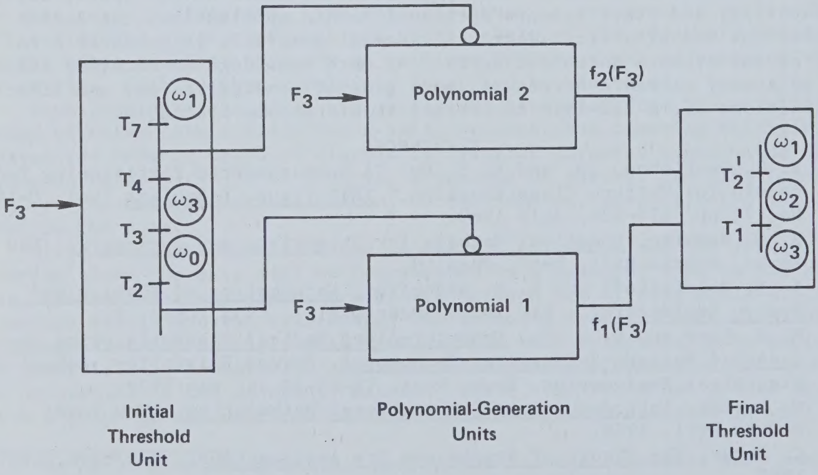

F_3
ω_1
T_7
T_4
ω_3
T_3
ω_0
T_2
F_3
Polynomial 2
$f_2(F_3)$
F_3
Polynomial 1
$f_1(F_3)$
ω_1
T_2'
ω_2
T_1'
ω_3
Initial Threshold Unit
Polynomial-Generation Units
Final Threshold Unit

FIG. 3. STRUCTURE OF THE MODIFIED BSU

$$\begin{array}{cccc} & \omega_0 & \omega_1 & \omega_2 \\ \omega_1 & 0 & X & \\ \omega_2 & | & || & X \\ \omega_3 & | & || & | \end{array}$$

FIG. 4. ADJACENCY MATRIX FOR THE EXAMPLE OF FIG. 1

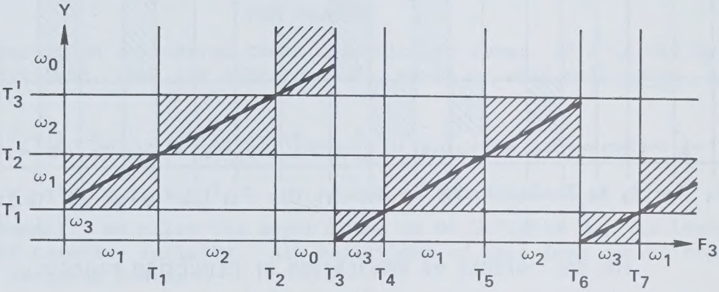

Y
ω0
T'3
ω2
T'2
ω1
T'1
ω3
ω1
ω2
ω0
ω3
ω1
ω2
ω3
ω1
F3
T1
T2
T3
T4
T5
T6
T7

FIG. 5. Y-AXIS ARRANGEMENT DETERMINED FROM ADJACENCY MATRIX

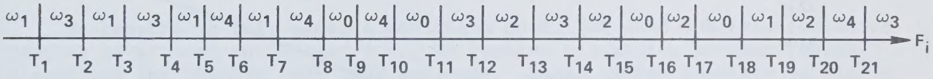

ω1 ω3 ω1 ω3 ω1 ω4 ω1 ω4 ω0 ω4 ω0 ω3 ω2 ω3 ω2 ω0 ω2 ω0 ω1 ω2 ω4 ω3
Fi
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18 T19 T20 T21

FIG. 6(a). CLASS LABEL STRING AND FEATURE-VALUE AXIS FOR EXAMPLE 4.1

$$\begin{array}{ccccc} & \omega_0 & \omega_1 & \omega_2 & \omega_3 \\ \omega_1 & | & X & & \\ \omega_2 & \enclose{circle}{|||} & | & X & \\ \omega_3 & | & \enclose{circle}{||||} & \enclose{circle}{|||} & X \\ \omega_4 & \enclose{circle}{|||} & ||| & | & | \end{array}$$

FIG. 6(b). THE CORRESPONDING ADJACENCY MATRIX

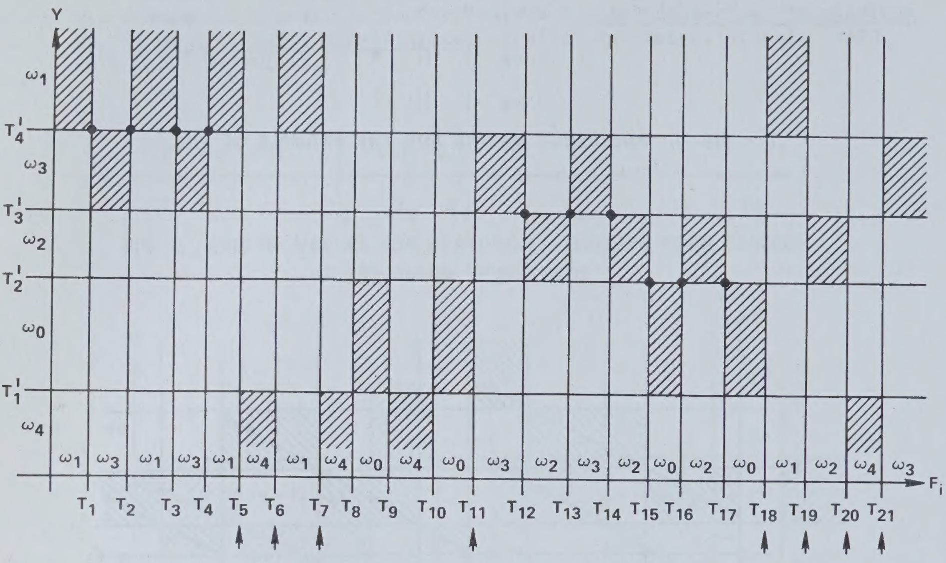

Y
ω_1
T_4'
ω_3
T_3'
ω_2
T_2'
ω_0
T_1'
ω_4
ω_1 ω_3 ω_1 ω_3 ω_1 ω_4 ω_1 ω_4 ω_0 ω_4 ω_0 ω_3 ω_2 ω_3 ω_2 ω_0 ω_2 ω_0 ω_1 ω_2 ω_4 ω_3
F_i
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19} T_{20} T_{21}

FIG. 6(c). RESULT OF APPLICATION OF REDUCTION PROCESS

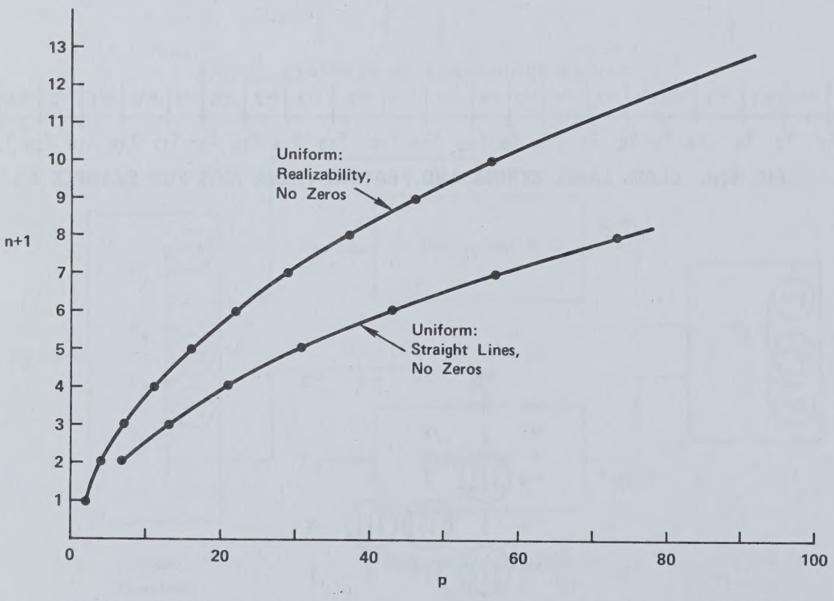


FIG. 7. CONDITIONS UNDER WHICH THE REDUCTION PROCESS WOULD BE ADVANTAGEOUS

A MODE ESTIMATING PROCEDURE AND AN OPTIMUM RECLUSTERING ALGORITHM

DEMETRIOS A. MICHALOPOULOS
Department of Quantitative Methods
California State University
Fullerton, California

ABSTRACT

The efficient partitioning of a finite-dimensional space by a decision tree, each node of which corresponds to a comparison involving a single variable; is a problem often occuring. This paper obtains a sufficient partition of the space based on labeled samples of each class (each partition is a mode). The optimum reclustering will be discussed in a future paper due to space limitations.

THE PROBLEM

The partition considered takes a particular form: if X is the space to be partitioned, then the subregions X_i, which we will call modes, are given by

$$X_i=\left\{\underline{x} \mid a_{i1} \leqslant x_1 \leqslant b_{i1},\ldots,a_{in} \leqslant x_n \leqslant b_{in}\right\},\ i = 1,2,\ldots,R \quad ,(1.1)$$

where

$\underline{x} = (x_1,x_2,\ldots,x_n)$. or unions of subregions X_i as above.

In general , we allow the upper bound to be infinity and the lower bound to be negative infinity. All partitions of this form can be represented by decision trees.

We will phrase most discussion in the remainder of this paper in terms of the pattern recognition problem. We are given samples of K classes,

$$\underline{y}_1^{(i)}, \underline{y}_2^{(i)}, \ldots, \underline{y}_{N_i}^{(i)}, \quad i = 1,2,\ldots,K \ .$$

where

$$\underline{y}_k^{(i)} = (y_{k_1}^{(i)}, y_{k_2}^{(i)}, \ldots, y_{k_n}^{(i)}) \ .$$

We require in this paper a partition such that each subregion contains samples of at most one class; i.e.e., we require a decision boundary which classifies the samples with 100% accuracy. It is always possible to find a decision tree which gives 100% classification if points of differing classes are not coincident, although such a decision boundary may be overly complex. We will discuss a technique for avoiding overly complex decision boundaries in a later paper, but will be content at this point to simply consider the problem stated.

FINDING A SUFFICIENT PARTITION

A sufficient partition is a tree (or, equivalently, a set of five-tuples) which classifies all samples correctly. Such a partition has boundaries at a finite set of values on each of n axes:

$$\underline{\alpha}_i = (\alpha_{i1},\alpha_{i2},\ldots,\alpha_{iM_i}), \quad i=1,2,\ldots,n,$$

with

$$\alpha_{i1} < \alpha_{i2} < \cdots < \alpha_{iM_i} \ .$$

These values are then sufficient to define a perfect partition of the labeled samples; they are used in the second stage to find a more efficient partition.

The procedure of the algorithms is to consider successively projections of the samples onto one axis. Because of this fact, we are always working with what amounts to one-dimensional samples, i.e., a finite set of scalar values with a class label:

$$z_1^{(i_1)} \leqslant z_2^{(i_2)} \leqslant \dots \leqslant z_N^{(i_{N'})} \quad , \qquad (2-1)$$

where i_k is the class membership of the kth value, and $i_k \in F$, where $F \subset \{1, 2, \dots, K\}$. Note that the values are labeled by their ordering on the real line, and hence, in general, value subscripts do not correspond to sample subscripts. Further, the number of such points in a given discussion need not be equal to the total number of samples; for, in the algorithm, we may be talking about subsets of the original set of labeled samples. Let the sample labels $(i_1, \dots, i_{n'})$ be chosen from a set F, which is a subset of the set of integers 1 through k. Let I be an interval on the axis under consideration, the axis upon which the values in (2-1) are defined.

Then we define

$\mu_k(I)$ = the number of samples of class k in the interval, i.e. the number of values in (2-1) which fall in I.

Definition: An exclusion interval I for a set of integers and for a set of values given by (2-1) is an interval I such that, for some $i \in I$, no $z_k^{(i_k)}$ in I has $i_k = i$. The class i is the class of exclusion for I.

Definition: A maximal interval I* for the set of values given in (2-1) is as exclusion interval maximizing

$$Q(I) = \sum_k \mu_k(I) \quad . \qquad (2-2)$$

An exclusion interval is hence an interval which does not have samples of all classes with labels in. Thus, Q(I), Eq. (2-2), counts the number of samples falling in the exclusion interval I, and the maximal interval is the interval containing the most points in a sequence which does not include at least one class of the classes under consideration. It can be shown that a maximal interval will contain only points whose class membership takes all values of J except the class of exclusion. That is, the number of classes represented in a maximal interval I* is equal to one less than the cardinality of J.

The axes are cycled in arbitrary order and each axis under consideration constitutes an iteration. For each iteration, the samples under consideration are the values in Eq. (2-1). A maximal interval is found and the values corresponding to endpoints of that interval, excluding $-\infty$ or $+\infty$, are added to the list of discrete values defining modes for the axis being considered. The samples used in each iteration are those remaining in a mode which has samples of more than one class. The algorithm terminates when there is no longer any such mode.

CONCLUSION

A new method for the estimation of modes in the pattern recognition problem has been discussed. Unfortunately, space limitations did not permit the complete coverage of the topics promised by the title. Further discussion of the above topics will be covered in a future paper.

REFERENCES

1. I.P. Bottlik, Personal Communication.
2. D.A. Michalopoulos, PhD Dissertation, USC, 1971.
3. W.S. Meisel, "Mathematical Pattern Recognition", Academic Press, 1971.
4. H. Andrews, "Mathematical Techniques in Pattern Recognition", in press.
5. E.Henrichon, and K.S. Fu, "A Nonparametric Partitioning Procedure for Pattern Classification," IEEE-C, 1969.

ON THE OPTIMAL PROPERTIES OF A PARTITION ALGORITHM

P. C. Yue and C. K. Wong
IBM Thomas J. Watson Research Center
Yorktown Heights, New York

Abstract: It is shown that a natural partitioning algorithm based on the knowledge of access probabilities is optimal in three specific storage applications. These applications include (1) pagination, (2) organization of a permanent store, and (3) disk space allocation. The use of Schur functions as an optimization technique is introduced.

I. INTRODUCTION

This paper is concerned with the following problem. Given a set of equal-sized items and their request probabilities, one wants to partition them into m groups so that the performance in retrieval is optimized according to some given criterion. Three specific applications to storage partition are considered.

The first one is the allocation of program sectors (or records in data files) to pages in a machine with virtual memory. The second one is the organization of permanent storage units for an archive system. The third one is the placement of records in secondary storage devices, e.g. placement into the cylinders of a disk. It will be shown that although the criterion of performance differs from application to application, there exists a natural partition algorithm which optimizes the performance in all situations.

Throughout this paper, we shall assume that there are items $R_1, R_2, \ldots, R_M$ with request probabilities $\alpha_1, \alpha_2, \ldots, \alpha_M$, $\sum_{i=1}^{M} \alpha_i = 1$, and the requests are independent random variables.

II. PRELIMINARIES

A unified approach to these problems is provided by the use of Schur functions. One would need the following definitions and result in order to prove optimality (see [6], p. 178).

Definition 1: If $q_1 \geq \cdots \geq q_m$, $p_1 \geq \cdots \geq p_m$, $\sum_{i=1}^{k} q_i \geq \sum_{i=1}^{k} p_i$ for $k = 1,2,\ldots,m-1$, and $\sum_{i=1}^{m} q_i = \sum_{i=1}^{m} p_i$, then $\underline{q} = (q_1,\ldots,q_m)$ is said to majorize $\underline{p} = (p_1,\ldots,p_m)$, written $\underline{q} \succ \underline{p}$.

Definition 2: A real function $\phi = \phi(x_1,\ldots,x_m)$ of m real variables is said to be a Schur function if for every pair $i \neq j$,

$$(x_i - x_j)\left(\frac{\partial \phi}{\partial x_i} - \frac{\partial \phi}{\partial x_j}\right) \geq 0$$

Theorem 1: Let ϕ be defined for $x_1 \geq \cdots \geq x_m$. Then $\phi(q_1,\ldots,q_m) \geq \phi(p_1,\ldots,p_m)$ for all $\underline{q} \succ \underline{p}$ if and only if ϕ is a Schur function.

The above theorem is due to Schur [7-8].

III. PAGINATION

The first problem considered is that of improving program behavior in a paging environment by improving locality. Suppose there are mk equal-sized relocatable program sectors (or records in the case of data files), $R_1, R_2, \ldots, R_{mk}$ with request probabilities $\alpha_1, \alpha_2, \ldots, \alpha_{mk}$, $\sum_{i=1}^{mk} \alpha_i = 1$, which are to reside in pages of k records each (i.e., a total of m pages). When requests are processed, the program sectors (or records) will be brought into the main memory in units of pages. The problem is to partition these mk sectors so that the subsequent processing of paging requests will be optimal. Two different measures of optimality will be employed.

Let the set of pages be enumerated by $\{1,2,\ldots,m\}$. Let $\{r_t\}$, $t = 0, \pm 1, \pm 2, \ldots$, be the sequence of page requests, i.e., r_t is the page referenced at time t and it takes value in the set $\{1,2,\ldots,m\}$.

The first measure of locality is defined as follows. Let N be a fixed positive integer and let $K_t^{(N)}$ be the number of distinct values of the set $\{r_{t-N+1},\ldots,r_{t-1},r_t\}$, i.e., the cardinality of the set. Then $K_t^{(N)}$ is a random variable. Since its distribution is independent of t, we shall write it as $K^{(N)}$. Its expectation $\overline{K^{(N)}}$ will be used as a measure of locality in the sequence of requests $\{r_t\}$.

The second measure is defined as follows. Suppose the request at time t is i and r_τ is the preceeding request for the same page i. Then the number of distinct values of the set $\{r_{\tau+1},\ldots,r_{t-1}\}$, denoted K is defined as the distance at time t, designated D_t. If this set is empty, K is defined as zero. D_t is a random variable with a distribution independent of t. It will be written as D and its expectation $\overline{D}$ will be the second measure of locality.

The first measure corresponds to the notion of working set size as discussed in [1-2]; the second corresponds to that of LRU distance used in [5] in connection with the study of stack processing. Both are indicators of the number of distinct pages referenced locally where "locally" in the latter case is defined by the interval between successive references to the same page while in the former case the interval is fixed but has variable length. These measures have the advantage of not involving the details of specific replacement algorithms since it is known that the memory reference pattern has a more profound effect on paging performance (see, e.g. [4]).

We shall describe the allocation scheme and prove that it is optimal in the sense of minimizing both $\overline{K^{(N)}}$ for all N and $\overline{D}$.

First, rearrange the probabilities of request in descending order: $\alpha_{\sigma^*(1)} \geq \alpha_{\sigma^*(2)} \geq \cdots \geq \alpha_{\sigma^*(mk)}$, where σ^* denotes the corresponding permutation of $\{1,2,\ldots,mk\}$. Then order the set of items according to $\sigma^*(1), \sigma^*(2), \ldots, \sigma^*(mk)$, and partition them into m groups of k items each in this order. Finally, each group is allocated to a page. For example, $R_{\sigma^*(1)},\ldots,R_{\sigma^*(k)}$ reside in the first page, say, and $R_{\sigma^*(k+1)},\ldots,R_{\sigma^*(2k)}$ in the second one, etc.

Note that each allocation scheme corresponds to a permutation σ of $\{1,2,\ldots,mk\}$ such that $R_{\sigma(1)},\ldots,R_{\sigma(k)}$ reside in the first page, $R_{\sigma(k+1)},\ldots,R_{\sigma(2k)}$ reside in the second page, etc. Since the numbering of pages and the ordering of items within a page are immaterial for the optimization of allocation schemes, one has only to consider the set Λ of all permutations σ of $\{1,2,\ldots,mk\}$ such that

$$\text{(i)}\quad \alpha_{\sigma((i-1)k+1)} \geq \cdots \geq \alpha_{\sigma(ik)}, \quad \text{for } i = 1,2,\ldots,m$$

and

$$\text{(ii)}\quad q_i^{(\sigma)} \geq q_{i+1}^{(\sigma)} \quad \text{for all } i,$$

where

$$q_i^{(\sigma)} = \alpha_{\sigma((i-1)k+1)} + \cdots + \alpha_{\sigma(ik)}.$$

Once σ is given, the i^{th} page will have request probability $q_i^{(\sigma)}$ and both $\overline{K^{(N)}}$ and $\overline{D}$ will be functions of $\{q_i^{(\sigma)}\}_{i = 1,2,\ldots,m}$ only. Thus, the problem reduces to the following:

Let $\underline{q}^{(\sigma)} = (q_1^{(\sigma)},\ldots,q_m^{(\sigma)})$ be the probability vector corresponding to $\sigma\epsilon\Lambda$. One has to show that $\underline{q}^{(\sigma^*)}$ minimizes both $\overline{K^{(N)}}$ and $\overline{D}$ among all $\underline{q}^{(\sigma)}$, $\sigma\epsilon\Lambda$.

A very important property of the partition σ^* follows from Definition 1 and the normalization (2) of σ in Λ. It is given in

Lemma 1: $\underline{q}^{(\sigma^*)} \succ \underline{q}^{(\sigma)}$ for all $\sigma\epsilon\Lambda$. Therefore, it suffices to show that for any probability vector $\underline{q} = (q_1,\ldots,q_m)$, $\underline{p} = (p_1,\ldots,p_m)$ such that $\underline{q} \succ \underline{p}$,

$$\overline{K^{(N)}}\,(\underline{q}) \leq \overline{K^{(N)}}\,(\underline{p}), \text{ for all } N \qquad \ldots \quad (1)$$

and

$$\overline{D}(\underline{q}) \leq \overline{D}(\underline{p}). \qquad \ldots \quad (2)$$

To prove (1), one notes that $\overline{K^{(N)}}\,(\underline{q}) = m - \sum_{i=1}^{m} (1-q_i)^N$. Hence one has only to verify that $\phi = \sum_{i=1}^{m} (1-q_i)^N$ is indeed a Schur function.

To establish (2), note that

$$\overline{D}(\underline{q}) = (m - \tfrac{1}{2}) - \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{q_i^2}{q_i + q_j}$$

Therefore, (2) is proved by showing that $\psi = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{q_i^2}{q_i + q_j}$ is a Schur function.

IV. ORGANIZATION OF A PERMANENT STORE

Here we are concerned with a slow but inexpensive large memory, for example, storage in a collection of tapes. Again, suppose there are mk records $R_1, R_2, \ldots, R_{mk}$ with request probabilities $\alpha_1, \alpha_2, \ldots, \alpha_{mk}$, $\sum_{i=1}^{mk} \alpha_i = 1$, which are to be partitioned into m groups of k records each. Each group is then placed on a tape. The entire collection of tapes is generally referred to as permanent storage.

Since not much is known in the literature about this type of application, only a hypothetical but feasible model is assumed. Suppose the requests are processed only when N of them have been accumulated, where N is a preassigned number. The corresponding set of tapes required to service these requests will then be mounted to read/write units for processing. This set of tapes is held in the read/write unit until the next N requests have been accumulated. Then whatever tapes required which are already in the unit will be retained and the rest will be returned to permanent storage. Those tapes which are required but not in the unit will be brought in accordingly. Processing then continues.

Suppose for each t, $S_t^{(N)}$ denotes the set of tapes required for the t^{th} period of N requests. The storage partition problem is then to find an assignment of the records to the tapes so that the expected size (cardinality) $L_t^{(N)}$ of the set $S_{t+1}^{(N)} - S_t^{(N)}$ is minimized. As in Section III, we shall consider the steady state. Let the sequence of request be $\{r_t\}$, $t = 0, \pm 1, \pm 2, \ldots$. The random variable $L_t^{(N)}$ has a stationary distribution with expectation $\overline{L^{(N)}}$. The subscript t will be suppressed in the sequel.

We shall show that $\underline{q}^{(\sigma^*)}$ as defined in III minimizes $\overline{L^{(N)}}$. For brevity, write $\underline{q} = \underline{q}^{(\sigma)}$, $\sigma \in \Lambda$. $\underline{q} = (q_1,,,,q_m)$ is now the vector of tape request probabilities. $\overline{L^{(N)}}$ can be expressed in terms of $\underline{q}$ as follows:

$$\overline{L^{(N)}} = \frac{1}{2}\,[m(m-1) - 2(m-1)\sum_{i=1}^{m}(1-q_i)^{2N} + \sum_{i\neq j}(1-q_i-q_j)^{2N}].$$

Let $\phi(\underline{q}) = 2(m-1)\sum_{i=1}^{m}(1-q_i)^{2N} - \sum_{i\neq j}(1-q_i-q_j)^{2N}$. It can be shown that $\phi(\underline{q})$ is a Schur function, hence by Lemma 1, $\underline{q}^{(\sigma^*)}$ minimizes $\overline{L^{(N)}}$.

V. DISK SPACE ALLOCATION

As before, suppose there are mk equal-sized records $R_1, R_2, \ldots, R_{mk}$

with request probabilities $\alpha_1, \alpha_2, \ldots, \alpha_{mk}$, $\sum_{i=1}^{mk} \alpha_i = 1$, which are to be placed in the m cylinders of a disk. Each cylinder is assumed to contain exactly k records. For convenience, we shall assume that m is odd. The objective is to minimize the expected access time to the cylinders.

We shall use the following matrix to characterize the access time:

$$C = [c_{ij}],\ 1 \leq i, j \leq m,$$

where c_{ij} represents the access time for cylinder j when the device is currently positioned at cylinder i, such that

(i) $c_{ij} = c_{ji}$,

(ii) c_{ij} is a function of $j - i$ only,

and

(iii) $j > k > i$ or $j < k < i$ implies $c_{ij} > c_{ik}$

Therefore, one has a two-stage problem:

(i) partition the records into m groups of k records each, and

(ii) assign each group to an appropriate cylinder, so that the expected cylinder access time is minimized.

We shall use the same notations and definitions of III, except that we shall call a page a group. Thus $\underline{q}^{(\sigma)} = (q_1^{(\sigma)}, \ldots, q_m^{(\sigma)})$ is the probability vector for the group requests when the set of mk records are partitioned according to $\sigma \epsilon \Lambda$, i.e., $q_i^{(\sigma)}$ is the request probability for the i^{th} group of records, $i = 1,2,\ldots,m$. Given $\underline{q}^{(\sigma)}$, the assignment of the groups to the m cylinders can be represented by a permutation π of $\{1,2,\ldots,m\}$. For example, the $\pi(i)^{th}$ group is placed in the i^{th} cylinder. Then the expected cylinder access time is given by

$$T(\sigma,\pi) = \sum_{i=1}^{m} \sum_{j=1}^{m} q_{\pi(i)}^{(\sigma)} q_{\pi(j)}^{(\sigma)} c_{ij}$$

From the symmetry of this function, it is clear that the assumption $c_{ij} = c_{ji}$ is only convenient but not necessary. Let Ω be the set of all permutations of $\{1,2,\ldots,m\}$. Then an optimal scheme would be one which minimizes $T(\sigma,\pi)$ over $\Lambda \times \Omega$.

Given $\sigma = \sigma_0$, it can be shown (see[3]) that $\pi^* \epsilon \Omega$ minimizes the function $T(\sigma_0,\pi)$ over Ω, where

$$\pi^*(1,2,\ldots,m) = (m, m-2, \ldots, 7, 5, 3, 1, 2, 4, 6, \ldots, m-3, m-1)$$

Recalling that $q_1^{(\sigma_0)} \geq \ldots \geq q_m^{(\sigma_0)}$, one notes that the group with the highest request probability is placed in the central cylinder, and successively less probable groups are placed adjacently on alternate sides. Let $T(\sigma_0,\pi^*) = T^*(\sigma_0)$. To minimize $T(\sigma,\pi)$ over $\Lambda \times \Omega$, it suffices to minimize $T^*(\sigma)$ over Λ. We shall show that the partition defined by σ^* in III indeed minimizes $T^*(\sigma)$ over Λ.

Therefore, an optimal scheme for problem III will be:

(i) use σ^* to partition the records into m groups.
(ii) order these groups in descending order of their request probabilities, and
(iii)use π^* to assign these groups to the cylinders of the disk.

By Lemma 1, all one needs to show is that $\underline{q}^{(\sigma)} \succ \underline{q}^{(\sigma')}$, $\sigma,\sigma' \in \Lambda$, implies $T^*(\sigma) \leq T^*(\sigma')$. For simplicity, write $\underline{q} = \underline{q}^{(\sigma)}$. Then $T^*(\sigma)$ is a function of $\underline{q}$, say $\phi(\underline{q})$. It can be shown that $-\phi(\underline{q})$ is a Schur function. Hence the optimality follows.

REFERENCES

[1] Aho, A., Denning, P., and Ullman, J. Principles of Optimal Page Replacement, J.ACM 18, 1 (January 1971), 80-93.

[2] Denning, P. The Working Set Model for Program Behavior, Comm. ACM 5, 11 (May 1968), 323-333.

[3] Grossman, D. and Silverman, H. F. Placement of Records on a Secondary Storage Device to Minimize Access Time, IBM Research Report RC 3641, December 1971. (To appear in Proc. of Sixth Annual Princeton Conference on Information Sciences and Systems, March 1972).

[4] Hatfield, D. and Gerald, J. Program Restructuring for Virtual Memory, IBM Systems Journal 10, 3(1971), 168-192.

[5] Mattson, R., Gecsei, J., Slutz, D., and Traiger, I. Evaluation Techniques for Storage Hierarchies, IBM Systems Journal 9, 2 (1970), 78-117.

[6] Marshall, A. W., Olkin, I., and Proschan, F. Monotonicity of Ratios of Means and Other Applications of Majorization, Chapter in Inequalities, Shisha (Ed.), Academic Press, New York, 1967.

[7] Ostrowski, A. Sur Quelques Applications des Fonctions Conveses et Concaves au sens de I. Schur, J. Math. Pures Appl. 31 (1952), 253-292.

[8] Schur, I. Über ein Klasse von Mittlebildungen mit Anwendungen auf die Determinatentheorie, Sitzber. Berl. Math. Ges. 22 (1923), 9-20.

PARALLEL GENERATION OF BINARY SEARCH TREES

C. K. WONG and S. K. CHANG
IBM Thomas J. Watson Research Center
Yorktown Heights, New York

Abstract: A method of constructing binary search trees in a multiprocessor computer system is proposed. To make better use of this method, balancing binary search trees is discussed.

I. INTRODUCTION

Binary search trees have long been recognized as useful structures for storing and retrieving data, particularly when sequential processing of data is also desired. A great deal of study has been devoted to the various aspects of this problem. Usually, the main concern is to minimize construction time as well as search time.

In many applications, records (data items) to be stored are not available simultaneously, but are received one by one in some arbitrary order. The tree is then constructed dynamically as the records come in. We shall assume that each record has a numerical key and the tree is to be constructed according to the values of the keys. For simplicity, we shall not distinguish between records and keys.

In this paper, we shall discuss such a dynamic construction of binary search trees in a multiprocessor computer system.

II. THE MODEL AND THE SCHEME

The following simplified model of a multiprocessor computer system is used. Assume that there are u identical processors, which can function independently and simultaneously. The computer memory consists of s identical modules, organized in a stack $M_1, M_2, \ldots, M_s$. Each module can be accessed by only one processor at a time, but different modules can be accessed by different processors simultaneously. If two processors have to access the same module at the same time, then one has to wait until the other finishes. On the other hand, in the input area, the processors form a queue. Each processor will fetch one input record, process it and go back to the end of the queue. We shall assume that except at the end of the process, at any given amount there will be at least one input record in the input area so that the first processor in the queue will never be idle.

At any instant, the tree being currently built is stored in the memory modules in the following manner. The record in the first level of the tree, i.e., the root, is placed in M_1. The records in the second level of the tree will be placed in M_2, and so on. The records in the $(s + 1)^{st}$ level will be placed back in M_1 again.

Records are to be inserted one by one into the tree. Each node in the tree has potentially two successors: a left successor and a right successor. The first record becomes the root of the tree. As a new record comes in, its key is compared to the key of the root node. If this new key is smaller, we pass to the left successor; if greater, to the right successor. (For convenience in analysis, the keys are assumed to be distinct.) The process is then repeated for the new node. If the appropriate successor does not exist, the new record is inserted as the missing successor, and the process terminates for that record.

Therefore, the first processor fetches the first record, puts it in M_1, which becomes the root of the tree, and goes back to the queue in the input area. The second processor fetches the second record, goes to M_1, compares it with its content, places it in M_2, and goes back to the queue. In general, a processor fetches a record in the input area, brings it to the stack of memory modules, walks down the tree, places it in the proper module, and goes back to the queue. For the input sequence 6, 7, 2, 4, 3, 1, 5, their locations in the modules will appear as follows (assume s = 3):

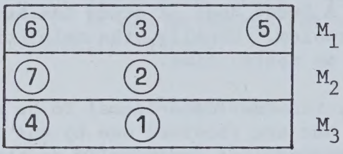


Figure 1.

III. THE TIMING

In order to estimate the construction time (per record) for a given input sequence, we assume that a comparison or placing the record in its proper place takes one unit of time.

Since we are interested in expected times only, some probabilistic assumption on the input sequence is necessary. For this purpose, we further assume that all permutations of the ranks of the input sequence are equally likely.

Let $E_{us}(t_n)$ be the expected time per record needed to construct the tree, where n is the number of records to be processed. Then

$$E_{us}(t_n) = \frac{1}{\tau}(1.4 \log_2 n) + O(\tfrac{1}{n}) \quad \ldots \quad (1)$$

where $\tau = \min(u,s)$, $O(\frac{1}{n})$ is a quantity which is bounded as $n \to \infty$.

The proof is as follows. For a given input sequence the resulting tree is completely determined, hence the level of the individual record in the tree is also determined. Note that the time needed to place a record in the tree is equal to its level. Let P_i be the level of the i^{th} record r_i, then $\lceil \frac{P_i}{s} \rceil$ is the number of times the record r_i visits M_1, where $\lceil x \rceil$ is the least integer greater than or equal to x.

Case (i) $u \geq s$

In this case, except for the very last phase of the process, at any given instant, there is always a processor visiting M_1. When r_1 visits (and stays in) M_1, it takes one unit of time. Then r_2 visits M_1, compares with r_1, another time unit passes. It next goes down to M_2, but at the same time r_3 comes in, visits M_1, compares with r_1. When r_3 goes down to M_2, r_4 visits M_1, and so on. Therefore, to count the total units of time elapsed, one has only to count the number of visits to M_1 by all the records $r_1, r_2, \ldots, r_n$, and then add in a correction term corresponding to the last s records. Hence the total time T satisfies the inequalities

$$\sum_{i=1}^{n} \left\lceil \frac{P_i}{s} \right\rceil \le T \le \sum_{i=1}^{n} \left\lceil \frac{P_i}{s} \right\rceil + \max_{n - s + 1 \le i \le n} P_i .$$

But

$$\frac{P_i}{s} \le \left\lceil \frac{P_i}{s} \right\rceil \le \frac{P_i}{s} + 1,$$

and $P_i \le n + 1$ for all i, hence

$$\sum_{i=1}^{n} \frac{P_i}{s} \le T \le \sum_{i=1}^{n} \frac{P_i}{s} + 2n + 1$$

Note that $\sum_{i=1}^{n} P_i$ is the total path length of the tree, whose expected value is equal to $n[1.4 \log_2 n + O(\frac{1}{n})]$. (See[2]). Equation (1) therefore follows.

Case (ii) $u < s$

Since increasing the number of memory modules will certainly not increase the total processing time, the timing of the present case will be less than or equal to that of the case when $s = u$, and the result follows.

IV. FURTHER IMPROVEMENT

In this section, we aim at reducing the factor 1.4 in (1) since it is a dominating factor in the timing. One way to achieve this is to balance the tree periodically. There are many ways to do so, the following one is proposed mainly for its conceptual and analytical simplicity.

First recall that a balanced search tree with depth ℓ can be simply defined as a search tree with $n = 2^{\ell+1} - 1$ nodes such that any path from the root to a terminal node contains exactly $\ell + 1$ nodes (including the root and the terminal node).

Consider the case of $u = 1$, $s = \infty$ first. Let ℓ be a preassigned positive integer. Construct a binary search tree for the first $n_1 = 2^{\ell+1} - 1$ records. Then rearrange them into a balanced search tree T_1. Using this balanced tree as a new starting point, insert the next $n_2 - n_1$ records into it, where $n_2 = 2^{2(\ell+1)} - 1$. Then rearrange the tree into a balanced one with n_2 nodes and of depth $2\ell + 1$. Denote it by T_2. Repeat the process again. In general, after k steps the resulting balanced tree T_k will have $n_k = 2^{k(\ell+1)} - 1$ nodes, and will be of depth $k\ell + k - 1$.

In order to compute the cost of construction, it is assumed that in the process of rearranging the nodes into a balanced tree, each node will take, on the average, $c \geq 0$ units of time. We also assume that the input sequence is infinite and for all n, all permutations of the ranks of the input sequence consisting of the first n records are equally likely.

Let E(1) be the expected construction time per record for T_1. To construct T_2, we have to insert $n_2 - n_1$ records into T_1. Each of these will go down T_1 from the root to a terminal noe and then will settle down in one of the left or right subtrees of this terminal node. Since there are 2^ℓ terminal nodes of T_1, there are $2^{\ell+1}$ such subtrees. The expected total time for the $n_2 - n_1$ records to form these $2^{\ell+1}$ subtrees after walking down T_1 is given by

$$2(n_2+1) \sum_{i=n_1+1}^{n_2} \frac{1}{i+1} - (n_2 - n_1).$$ (See [1], Theorem 1.) Therefore,

the expected total time to construct T_2 is given by $n_2E(2) = n_1E(1) +$

$$(n_2 - n_1)\ (\ell + 1) + 2(n_2 + 1) \sum_{i=n_1+1}^{n_2} \frac{1}{i+1} - (n_2 - n_1) + cn_2.$$

In general, for $k \geq 2$, the expected time to construct T_k is given by

$$n_kE(k) = n_{k-1}E(k-1) + (n_k-n_{k-1})(k-1)(\ell+1) + 2(n_k+1) \sum_{i=n_{k-1}+1}^{n_k} \frac{1}{i+1} - (n_k-n_{k-1})$$
$$+ cn_k.$$

Neglecting small terms, one has

$$E(k) = (bk + d)\left(\frac{a^{k-1}-1}{a-1}\right) - b\left[\frac{(k-1)a^k-ka^{k-1}+1}{(a-1)^2}\right] + a^{k-1}E(1),$$

where $a = \frac{1}{2^{\ell+1}}$, $b = (1 - a)(\ell + 1)$ and $d = 1.4(\ell + 1) + c - (1 - a)$.

For large ℓ one has

$$E(k) = \log_2 n_k + 0.4(\ell + 1) + c - 1 \qquad \ldots \qquad (2)$$

We are ready to discuss the general case. The construction scheme is exactly as described in II except that when the number of records reaches n_1, n_2,..., a balancing is in order. At that moment the records are rearranged and put into appropriate modules according to levels. For example, for the input sequence 6, 7, 2, 4, 3, 1, 5, the record locations before balancing are shown in Figure 1. The configuration after balancing is as follows:

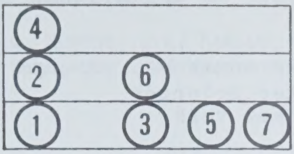


Figure 2.

To estimate the timing, let the average time to rearrange a record be θ_{us}, then $\frac{c}{\tau} \leq \theta_{us} \leq c$. By the same argument as before one can show that the expected time per record to construct a tree of size n_k is given by

$$E_{us}(k) = \frac{1}{\tau} \; [\log_2 n_k + 0.4(\ell + 1) - 1] + O\left(\frac{1}{n_k}\right) + \theta_{us}$$

$$= \frac{1}{\tau} \log_2 n_k + O\left(\frac{1}{n_k}\right) \qquad \ldots \qquad (3)$$

Hence, the factor 1.4 in (1) is eliminated.

V. CONCLUDING REMARKS

So far we have discussed the case when the parameter ℓ is preassigned and fixed. The optimal choice of ℓ certainly depends on the system and the size of the file involved. Therefore, it should be decided upon from case to case. Another variation might be a dynamically adjusted parameter. For example, a threshold can be set so that if the currently built tree has a total path length exceeding that of the corresponding balanced tree plus the threshold value, a balancing is invoked.

The purpose of this paper is to demonstrate how one can effectively construct binary trees in a multiprocessor computer system. Due to the balanced configuration of the tree, subsequent search would be faciliated. Furthermore, these methods can also be used for sorting a file.

REFERENCES

[1] Frazer, W. D. and McKellar, A. C., "Samplesort: A Sampling Approach to Minimal Storage Tree Sorting", J.ACM, Vol. 17, 3, July 1970, pp. 496-507.

[2] Hibbard, T., "Some Combinatorial Properties of Certain Trees", J.ACM, Vol. 9, 1962, p. 13.

[3] Nievergelt, J. and Wong, C. K., "On Binary Search Trees", Proc. of IFIP Congress 71, Ljubljana, Yugoslavia, August 23-28, 1971.

A MARKOV ALGORITHM COMPUTER: AN EDUCATIONAL TOOL FOR COMPUTABILITY

MARTIN E. KALISKI AND NORMAN R. KASHDAN
Department of Computer Sciences
The City College of New York
New York, New York

This presentation summarizes an interactive computer system to program and test Markov algorithms [1,2]. The motivation for and a brief description of this system is given below.

Many of the intuitive notions of computability are reflected in a student's ability to develop his own Markov algorithms. Unless a student can actually test his developed schema he is in the unfortunate position of writing a complex algorithm which he must assume works, but which is largely untestable due to the limitations of hand testing.

A Markov algorithm computing system is, consequently, being developed to allow the user to specify the productions of his algorithm in a natural way and to test its operation for any of a large class of test strings that the user also provides. The system can be instructed to print the results of each intermediate computation of the algorithm in the processing of the string or to simply print out the final result. Through its interactive nature the student can modify his production schema in an on-line fashion and through the file system of the computer he can call upon already written programs to build up complex algorithms out of simpler ones.

The system has educational potential as a tool for illustrating various abstract computational concepts to the beginning student and its implementation is part of an overall plan of the authors to design a more general system which incorporates interactively different mechanical models of computability,e.g. Turing machines or Kleene equations. Thus, this work is in the spirit of current efforts in the field of computer-aided instruction [3].

References: 1. Mendelson, E. Introduction to Mathematical Logic, Van Nostrand, 1964.
2. Kashdan, N. A "Markov Algorithm Computer", submitted in course CS 5729, The City College of New York, May, 1972.
3. Pressman, I. Computer-Assisted Instruction: A Survey, IEEE Transactions on Education, Vol. E-13, No.3, September, 1970.

A TELEGRAPHICS SYSTEM FOR THE ARDS-100A* (*ADVANCED REMOTE DISPLAY STATION)

P.J. CAUDILL, R.G. NESTOR, C.A. HARLOW, AND S.J. DWYER,III
Dept. of Elect. Eng., Univ. of Missouri-Columbia, Columbia, Missouri

ABSTRACT

The report describes the two year effort of the Image Analysis Laboratory Staff to develop an interactive graphics software package on an IBM 360 for use with the ARDS (Advanced Remote Display Station) terminal. The current version of OASIS (Optimal ARDS Software Interface System) provides three forms of support to the user. Section one of the package is the simulator, used to aid the programmer in debugging his software in a batch environment. Written in FORTRAN IV, it uses the line printer to simulate the ARDS output screen. Programs written and debugged under the simulator will run without modification on ARDS/PAK (Section two of OASIS). ARDS/PAK is written in FORTRAN IV and IBM 360 Assembler (BAL) for use under the Baylor Executive System for Teleprocessing. Section three of OASIS is a CALCOMP Plotter - ARDS interface which allows a user to run his CALCOMP programs on the ARDS with minor modifications. The entire OASIS package was written in Assembler and FORTRAN IV and runs in 60K+ bytes, depending on the size of the user program. A programmer can typically begin writing programs after as little as a two hour introduction to OASIS. User programs may be written in Assembler, FORTRAN, PL/1, or COBOL as standard IBM linkage conventions are used throughout the package. Release II of OASIS has been expanded to provide the user with some additional utility functions. These include data conversion routines, extended error handling facilities, simulation of function keys and light pens, and scratch-pad storage management. Programs may address the ARDS screen in ARDS units (absolute addressing) or user units (relative addressing) depending on which OASIS routines are called. Relative addressing allows the user to define screen size and position of the addressing grid origin. The user may switch between addressing modes or mix them depending on his application. A complete description of OASIS including a User's Guide with sample programs is available in Technical Report IAL-TR 7-72 from the Image Laboratory, Department of Electrical Engineering, University of Missouri-Columbia, Columbia, Missouri.

ARDS

Computer Displays Advanced Remote Display Station (ARDS) is an important innovation in computer terminals. It couples third generation electronics with a new display technology that allows a complete graphic station to be built into a self-contained, desk-top console. An almost unlimited amount of complex graphics as well as alphanumeric symbols can be drawn on ARDS with remarkably precise detail yet absolutely free of flicker or drift.

ARDS is designed to communicate with any computer, either directly connected, or over a standard telephone line. ARDS interface is extremely simple. To the computer it looks like a teletypewriter except that it can communicate data at 1200 bits per second (or faster). ARDS transmits and receives data in character asynchronous, bit-serial form. The code used is ASCII.

The basic ARDS unit logically and physically consists of three parts: controller, display unit, and keyboard (Figure 1).

The display unit contains a direct-view storage CRT. The surface of the display unit acts as the console's memory, and serves the double purpose of image storage and viewing screen. Once data is written on the surface of the display unit, it remains visible for more than 15 minutes without noticeable degradation. To clear the viewing surface for fresh data, the screen of the display unit erases in one-half second. The display can also present a "non-storing" cursor, which can be moved about on the screen without leaving a trace behind it. This is useful for locating beam position and for graphic input.

In normal operation the keyboard provides alphanumeric input to the computer and to the display unit. It contains keys for generating all 128 possible codes of ASCII. Patterned after a normal typewriter, the keyboard contains a pair of shift keys, and shift lock.

The controller of ARDS contains the digital logic of the system. In it is the electronics associated with the computer interface, the code conversion and analog signal generation. It contains a symbol generator and a vector generator, which permit extraction of the fullest information transmission capacity of a narrow band communication link to the computer.

The controller also contains the logic for formatting and transmitting the data from the keyboard into the computer. The input section and the output section of the controller are functionally separate, although data transmitted from the keyboard can be fed back locally to the output section. The viewing area of the display unit is 8 1/4" x 6 3/8". Spots plotted on 20-mil centers will appear as discrete dots. With the excellent resolution of display unit, it is possible to plot over 4,000 easily legible symbols.

For graphics, the display screen is defined to contain 1081 x 1415 addressable points (these are not necessarily resolvable points). The vertical edges have x addresses of +540 and -540, and the horizontal edges have y addresses of +707 and -707. The point 0, 0 is in the center of the screen. This addressing type is referred to as ARDS-Units in the OASIS description. No special precautions must be taken if the user's total picture is somewhat larger than the viewing window of the display. The transmission rate of the line limits the writing rate.

The ARDS Keyboard contains 58 keys and is capable of transmitting all 128 ASCII codes.

There is no character storage in ARDS so that each key, as it is struck, transmits the associated character to the computer. The characters transmitted may also be coupled back within the controller to the output printing section of ARDS, so that striking a key will cause the symbol to appear on the screen. This local "echo" is under program control. Most keys are normal printing characters or are standard carriage controls. Shift causes upper case characters to be transmitted. The Erase key sends the FF (Form Feed) characters. If the output (printing) section of ARDS receives FF, it erases the screen and moves the beam to the upper left hand corner of the screen.

The controller can be considered to have two sections: input and output. The input section, along with the keyboard and Graphic Input Option, creates, formats and transmits data to the computer. The output section with the display receives, decodes and prints data from the computer. It contains the symbol generator, vector generator, and serial-to-parallel buffers for creating pictures on the display. The controller can be put into either of two modes: symbol or graphic. In symbol mode, ARDS acts like a normal teletypewriter. In graphic mode, ARDS interprets characters, not as symbols to be plotted, but as binary data for drawing lines or plotting points.

The symbol generator set contains the 96 printable symbols (counting Space and Delete) of the American Standard Code for Information Interchange (ASCII).

Although symbol size can be adjusted to the user's requirements, the standard size is 12 characters per inch. This permits 80 symbols per line and 50 lines to be displayed.

In symbol mode ARDS will act as any teletypewriter, plotting any printing character received and moving the beam to the right one symbol position.

In graphic mode ARDS can position the beam in absolute coordinates. The command to position the beam, called a Set Point command, specifies the coordinate values as sign-magnitude numbers. Magnitudes for x and y are 10 bits each.

The vector generator also operates on sign-magnitude numbers and draws vectors relative (Δx, Δy) to the last beam position. Magnitude values for Δx and Δy are 10 bits each and thus lines can have x and y components up to 1023 increments, which is full screen width.

Vectors can be blank, intensified, or dotted under program control.

The data format of ARDS is compatible with ASCII, the symbol set contains the 96 printable ASCII symbols. To accommodate graphic input and output, ARDS operates under mode control. ARDS is set into symbol mode by any of the ASCII control characters with the exception of three that are reserved for entering into graphic mode.

Since ASCII does not have any provision for sending "graphic" information, a scheme for extending the code for ARDS has been adopted. This extension, which does not violate the basic precepts of ASCII, provides for a large number of "graphic commands," each of which can interpret "binary" characters as arguments Most of these commands are not assigned and are available for future extensions. The standard ARDS responds to three graphic commands as indicated below:

1. Set Point will locate the beam to any absolute location on the screen.
2. Long Vector will draw a relative vector any length up to 1023 increments in any direction, blanked, or visible; solid or dotted.
3. Short Vector will draw a relative vector in any direction any length up to 31 increments, always visible and solid.

Once in graphic mode ARDS looks for "binary" characters to be arguments of the mode, e.g. Set Point, Long Vector, or Short Vector. They contain just 6 bits of information. By this assignment, binary data never looks like communications control characters which may cause undersired actions along the communications line. ARDS will stay in a mode until it receives a character that takes it to another mode or back to symbol mode.

As the ARDS looks like a teletype to the computer, it may be used with most teleprocessing systems. We are currently using the Baylor Executive System for Teleprocessing on a dial-up line to an IBM 360/50.

Many of the ARDS control characters are not normally valid, and many normally valid control characters are not recognized by the display device. This effect has been minimized by inserting a special translate table into Baylor.

To provide uniformity throughout the software system, conversion between EBCDIC and the ARDS version of ASCII is done immediately prior to transmission and immediately after reception. In this manner, all data stored internally is in EBCDIC, the standard internal code of the IBM 360.

Character messages and graphics command chains are assembled into one hundred twenty byte buffers and passed to the teleprocessor by OASIS. There, it is enqued in a separate buffer on direct access storage to await transmission. When Baylor finds the line free, it deques the message and shoves it down the appropriate I/O channel.

Input is achieved in the reverse of this process. When a message comes down the line, Baylor enques the message on disk. As data is needed, OASIS asks for this message. Baylor then deques the first transmission available and returns it to the calling program where it is broken down and given to the problem program.

The ARDS Graphic Support Package, OASIS, developed at the University of Missouri-Columbia, is a collection of subroutines which allow the programmer to perform basic telegraphic operations using the ARDS 100-A. In the process of operating any peripheral equipment it becomes advantageous to use some form of access method to provide the basic operating function for the programmer. In the normal processing of programs under OS/360, the programmer uses access methods without realizing it. For example, in doing input-output one might use BSAM, QSAM, BPAM or BDAM. In doing telecommunication one might use BTAM or QTAM. The purpose of this software interface is to allow the programmer a simple subroutine call to accomplish his purpose. Although OASIS is not an access method, it works in a like manner by allowing a simple call from the user's program to accomplish telegraphics.

This collection of subroutines is basically designed to be called from Fortran or Assembler. However, PL/1 linkage is provided and OASIS may be used from a Fortran, PL/1 Assembler or any high level language using Fortran-like linkage such as COBOL. Research development now leads to use of OASIS with the language LISP.

OASIS has been so designed as to optimize the use of the ARDS over low speed telecommunication lines. OASIS provides automatic mode switching, buffer segment management, and the generation of ARDS hardware instructions from programmer-oriented data. OASIS also allows use of the ARDS in a programmer defined coordinate reference frame, in program defined units, and in ARDS raster units. As a graphic input device, the OASIS will automatically decode both set point and long vector instructions from the ARDS as can be generated by use of either the mouse of joystick. OASIS provides automatic facilities for LOG ON and LOG OFF of the telecommunication message switching system being used.

The basic ARDS Package provides facilities to draw lines in any of the modes available to the ARDS. There is an optional tracking feature which allows lines to be drawn by specifying the endpoints. While providing a basis for ARDS programming, this feature does not make it convenient for programmer use; therefore, OASIS provides a higher level of support, the user-defined Cartesian coordinate system. This level in the package is an automatic conversion from user-defined Cartesian coordinates into ARDS raster units. Automatic scissoring of pictures is provided. Many of the more drawing functions are automatically supplied, e.g. polygons, function plots, scatter diagrams of user data. These routines were kept as simple to use as possible; in particular, they have not been "over-parameterized".

Some of the other advanced functions which are available to the ARDS/ OASIS user are described below. They include sections for simulating light pens, function keys and a user scratch pad work area management routine (Figure 2).

ROTATION ROUTINES

TOTAT2 and ROTAT3 are two subroutines designed to rotate two and three dimensional data respectively. These two routines are written in Fortran IV, thus making them readily adaptable to most software.

These two routines will rotate data around each of the coordinate axes. The user's data may be rotated about the origin, a point or an axis. This rotation scheme is a pointwise manipulation of matrixes applied to the geometry of a coordinate system.

AREA STORAGE

In the course of programming, it is worthwhile to store certain things. The feature is provided for on the ARDS by OASIS. It provides the user with the ability to dynamically allocate blocks of storage regardless of the user's programming language. Each allocation is uniquely identified by a user-supplied number. One interesting use of these areas is to build Macro elements of command strings. For example, in circuit design the user could construct the commands necessary to display a resistor, and then recall the composite command string to generate a resistor on the ARDS screen.

SIMULATED FUNCTION KEY FEATURE

During the execution of a program it sometimes becomes advantageous to branch to a subroutine to handle a special function. Most graphics systems have a provision for function keys. This is not true on the ARDS, so a simulation feature has been added. This feature is used via a call to a set of Assembler routines. The routine AKEY defines a list of subroutines to be used during the execution of a function key simulation. The subroutine names passed in the parameter list to AKEY define a table where the first entry corresponds to function key number 1, the second to function key number 2, etc. Entries in this table may be deactivated and reactivated at any time by a call to ADQKEY and AEQKEY respectively, with the key number to be deactivated or reactivated as the parameter.

If an execution of an inactive function key is attempted, an error condition arises, and a call to the error condition routine is made.

In order to activate the function key simulator, a call to the routine AKEYIN must be made.

Execution is placed in a wait state until an intervention from the ARDS is made by keying in "**FKXX", where XX is the function key to be executed. For example, "**FK01" or "FK1" could be equivalent ways to execute function key subroutine number one. At the ARDS the message "KEYON-"

will appear on the lowest line of the ARDS screen; this notifies the ARDS user that a function key is impending. For example, "KEYON-**FK1". The maximum number of function keys permitted is an installation-defined parameter in the assembly of the simulator routine, set now at thirty-one function keys.

SIMULATED LIGHT PEN FEATURE

Although the ARDS does not have a light pen, per se, it does have two features which are perhaps more powerful than a light pen. These are the set point and long vector commands generated by use of the mouse or joystick. The subroutines AGETPT and UGETPT automatically provide a means of decoding these instructions directly, giving either a point or line segment. At times it becomes advantageous to be able to detect a point anywhere within an area rather than a single point. This feature is provided by the light pen simulation routine ALTPEN. In order to utilize this feature, the regions must first be defined. These regions are defined in terms of ARDS raster units and the ARDS display screen. These areas may be intersecting or disjoint. For non-disjoint areas the order of priority is given by the order of definition. Any set point command generated within an area when the simulator is invoked will generate a call to a programmer-defined subroutine.

EXTENDED ERROR HANDLER FACILITIES

One extension of OASIS is the extended error handling facilities. This provision allows the user to intercept all non-system errors and perform his own error tracing or printing of diagnostics. The user provides OASIS with the name of a subroutine to be executed in the event of an error. In this subroutine the user is further given the opportunity to do his own error analysis and/or request the OASIS system to take standard corrective action.

THE STRUCTURE OF THE CALCOMP-ARDS INTERFACE

The CALCOMP-ARDS interface allows users of the CALCOMP device to run their CALCOMP programs on the ARDS simulator or the actual ARDS itself.

There are two primary routines in the CALCOMP-ARDS interface. These are subroutines "PAGE" and "PLOT".

Subroutine "PAGE" receives the limits of the user's data and sets the scaling factors to convert to ARDS raster units. This routine also initializes the ARDS software interface

Subroutine "PLOT" takes the received CALCOMP graphics commands and converts them to ARDS graphics commands. This routine also closes the ARDS software interface upon receiving a value of "999" for the argument of "IPEN".

In summary, the CALCOMP-ARDS interface provides a CALCOMP user to run his programs on the ARDS with no changes. There are four more modified routines in the CALCOMP-ARDS interface and they perform the same functions as the original routines but in ARDS commands. These routines are:

1. PLOTS
2. OFFSET
3. WHERE
4. FACTOR

THE LOGIC OF THE ARDS SIMULATOR

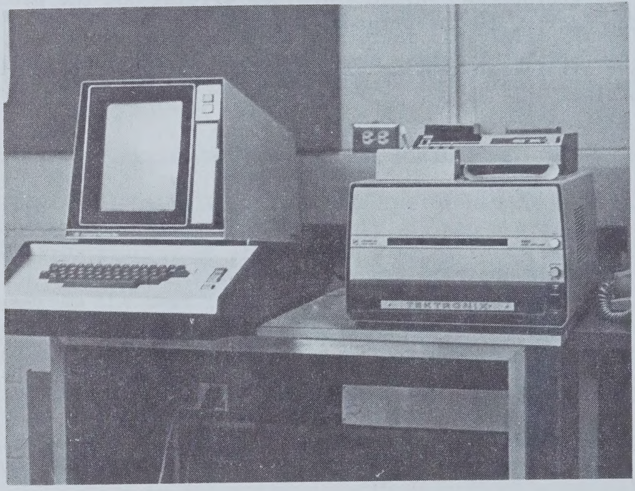
The ARDS simulator is designed to simulate teleprocessing on a system printer. The advantage of simulating teleprocessing on a printer is the time saved in debugging a teleprocessing program.

Since the ARDS software interface to perform teleprocessing only uses one subroutine, "AWTR", to perform input-output between the software interface and the operating system, the task of "rerouting" the output from a teleprocessing device to the printer is rather straightforward.

Subroutine "AWTR" receives a buffer segment of the output of the ARDS software interface. This routine decodes the output. AWTR scales the output and places appropriate characters in a 76 by 60 matrix used to simulate the ARDS screen.

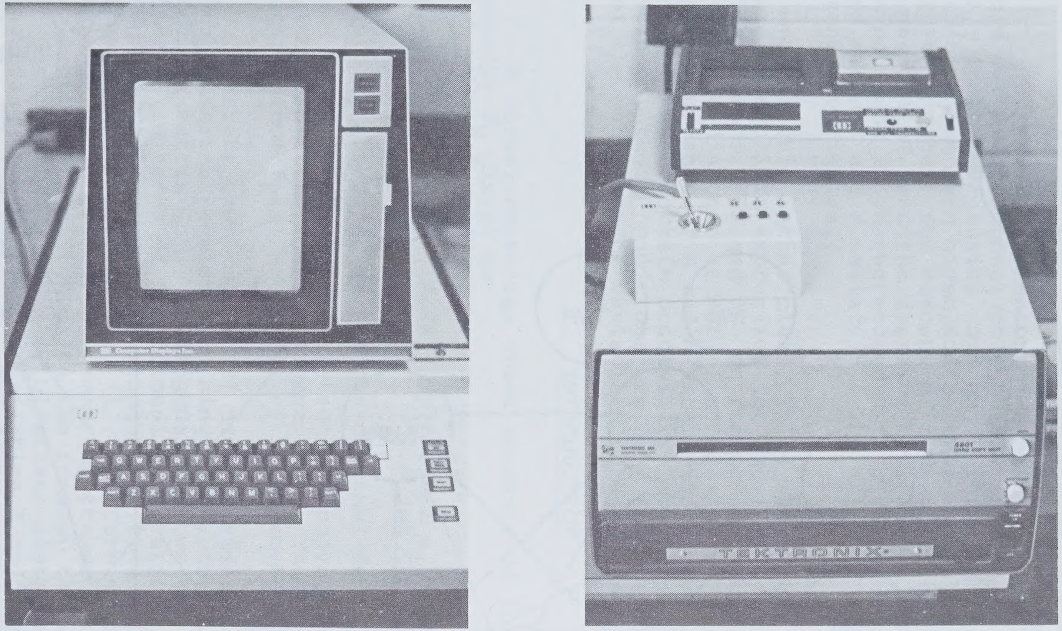
Because the user determines when he desires to have a screen of data displayed, the subroutine "PUTPIC" provides writing out the accumulated output in the matrix, to the printer.

An input simulation routine allows an interactive program to be tested on the simulator.



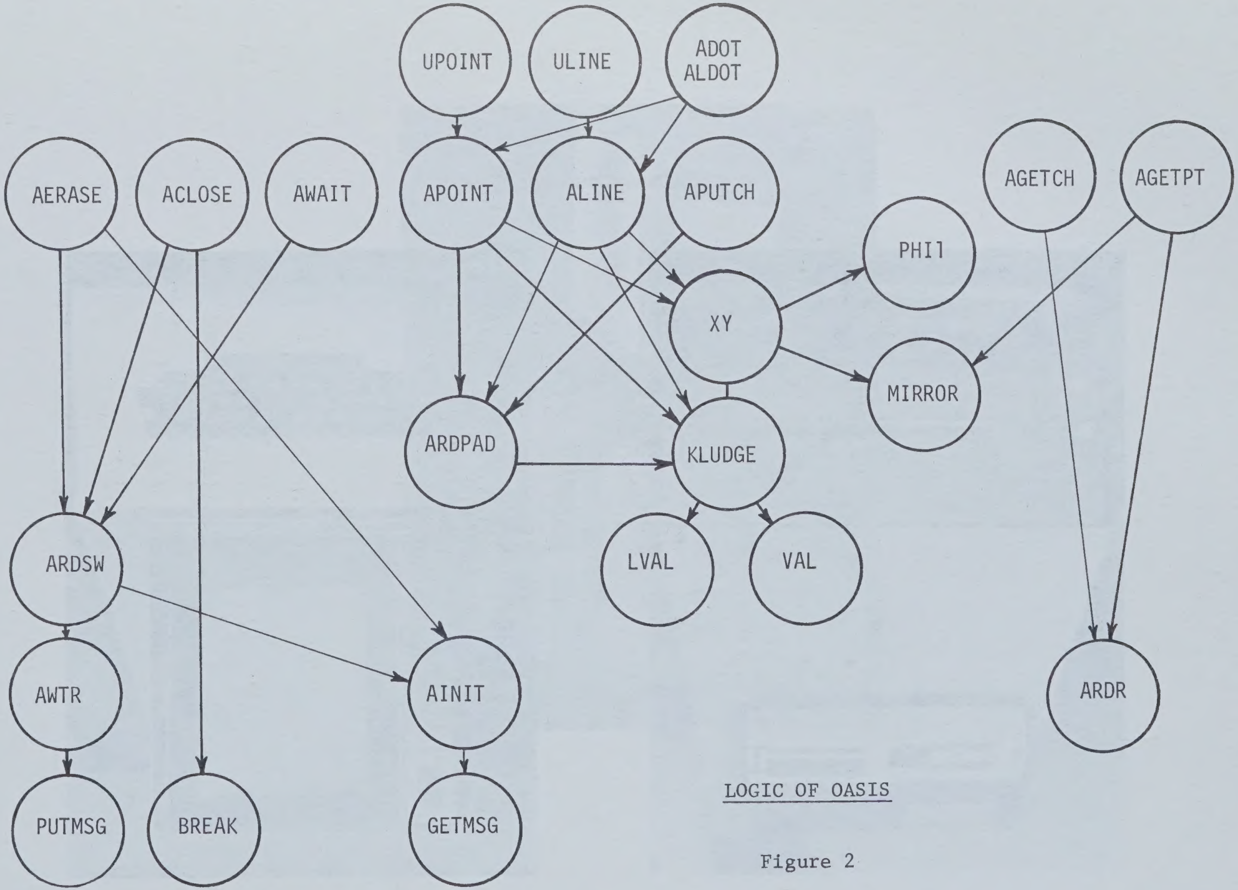
ARDS 100A and Installation

Figure 1a



ARDS 100A and Installation

Figures 1b & 1c


UPOINT
ULINE
ADOT
ALDOT
AERASE
ACLOSE
AWAIT
APOINT
ALINE
APUTCH
AGETCH
AGETPT
PHI1
XY
MIRROR
ARDPAD
KLUDGE
LVAL
VAL
ARDSW
AWTR
AINIT
ARDR
PUTMSG
BREAK
GETMSG

LOGIC OF OASIS

Figure 2

ON THE REPRESENTATION OF BIVARIATE DISTRIBUTIONS

HALUK DERIN AND JOHN B. THOMAS
Department of Electrical Engineering
Princeton University, Princeton, N.J. 08540

ABSTRACT

The series expansion of a bivariate destribution in a double set of orthogonal polynomials is well known. The coefficients ρ_{ij} in this expansion form an infinite matrix $[\rho_{ij}]$. Much attention has been given to the case where this matrix is diagonal. In this paper, the concept of a diagonal coefficient matrix is generalized, and the class of bivariate distributions with such expansions is characterized. In particular, the class is treated where X and Y satisfy Y = X + N, and X and N are independent random variables.

For this latter class, it is shown that $[\rho_{ij}]$ is triangular, and an explicit expression for $[\rho_{ij}]$ is determined in terms of the moments of X and N. On requiring that this expression for $[\rho_{ij}]$ be diagonal, a set of necessary and sufficient conditions on the moments of X and N is obtained. For X and N distributed according to certain probability laws, e.g. Gaussian, Gamma and Poisson, these conditions are used to determine the distribution of the other random variable N or X so that $[\rho_{ij}]$ is diagonal.

I. INTRODUCTION

A formal expansion often encountered [1] for the bivariate density $f_{X,Y}(x,y)$ is

$$f_{X,Y}(x,y) = f_X(x)f_Y(y) \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \rho_{ij}\varphi_i(x)\,\theta_j(y) \qquad (1)$$

where $f_X(x)$ and $f_Y(y)$ are the marginal densities of $f_{X,Y}(x,y)$ and $\{\varphi_i(x)\}$ and $\{\theta_j(y)\}$ are sets of polynomials orthonormal with respect to $f_X(y)$, respectively. These polynomials are generated from $f_X(x)$ and $f_Y(y)$ by the Gram-Schmidt procedure.

Barrett and Lampard [2] have considered the class where the double series reduces to a single one; that is, where the coefficient matrix $[\rho_{ij}]$ is diagonal. Brown [3] characterized this class Λ in terms of conditional expectations and later Wong and Thomas [4] in terms of the solutions to a Fokker-Planck equation. In most problems using Eq. (1), the form of the coefficient matrix $[\rho_{ij}]$ is clearly of great importance. In Section II, a collection of conditions on the form of this matrix is developed. In Section III, a class of bivariate distributions is defined by requiring that the random variables X and Y be interrelated by

$$Y = X + N \qquad (2)$$

where X and N are independent random variables. This restriction is common in estimation and detection theory and in the study of optimal systems. By imposing a set of necessary and sufficient conditions for the bivariate distribution of X and Y of Eq. (2) to have a diagonal expansion with polynomials, we define the proper subset Λ_o of the class Λ. This class Λ_o of bivariate distributions is characterized by a set of necessary and sufficient conditions involving the moments of X and N. By another approach, a different set of necessary conditions on the moments for class Λ_o is also determined.

II. FORMS OF THE EXPANSION

In the usual probability space setting, let $F_{X,Y}(x,y)$ be the bivariate distribution function and let $F_X(x)$ and $F_Y(y)$ be the marginal distribution functions. Also, let $\{\varphi_i(x)\}$ and $\{\theta_j(y)\}$ be orthonormal and complete sets of functions in the real Hilbert spaces $L^2(dF_X)$ and $L^2(dF_Y)$ respectively so that

$$\int_{-\infty}^{\infty} \varphi_i(x)\varphi_j(x)dF_X(x) = \delta_{ij} \tag{3}$$

and

$$\int_{-\infty}^{\infty} \theta_i(y)\theta_j(y)dF_Y(y) = \delta_{ij} \tag{4}$$

We assume well-behaved continuous random variables so that the density functions $f_{X,Y}$, f_X, and f_Y exist. Then the coefficient ρ_{ij} of Eq. (1) is

$$\rho_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \varphi_i(x)\theta_j(y)dF_{X,Y}(x,y) \tag{5}$$

and Eq. (1) holds in $L^2(m^2)$ where m is the Lebesque measure.

We are interested in expansions where the orthonormal functions $\{\varphi_i(x)\}$ and $\{\theta_j(y)\}$ are the polynomials generated by the Gram-Schmidt orthogonalization procedure from $1,x,x^2,x^3,\ldots$ and $1,y,y^2,y^3,\ldots$ using $f_X(x)$ and $f_Y(y)$ as the weighting functions, respectively. Despite the ease in obtaining these sets of orthogonal polynomials, they may fail to be a complete system. As Akhiezer and Glazman [5] assert, if the interval of orthogonality is finite (i.e. if the distribution function is constant outside a finite interval), then the orthonormal polynomials form a complete system. However if the interval of orthogonality is infinite, then the system of orthonormal polynomials may fail to be complete. Picone's theorem and other results on the necessary and sufficient conditions for the completeness of orthogonal systems are given in Sansone [6]. Completeness of such orthonormal polynomials is an important problem but we do not pursue it further here. Instead we assume that all conditions are satisfied for a bivariate distribution to be expressible as in Eq. (1), and we investigate various forms of the matrix $[\rho_{ij}]$.

We begin by listing some observations and conditions on the matrix $[\rho_{ij}]$:

Remark 1. Since $\varphi_0(x)$ and $\theta_0(y)$ are each unity, it follows from Eq. (5) that $\rho_{i0} = \rho_{0i} = 0$, $\forall i>0$, and $\rho_{00} = 1$.

Remark 2. $$E\{\varphi_i(X)|Y = y\} = \sum_{\ell=0}^{\infty} k_{i\ell}\theta_\ell(y) \tag{6}$$

if and only if $\rho_{ij} = k_{ij}$, $\forall j\geq 0$; $\forall i\geq 0$.

Proof: The coefficient ρ_{ij} may be written as

$$\rho_{ij} = \int_{-\infty}^{\infty} \theta_j(y)\left[\int_{-\infty}^{\infty} \varphi_i(x)\,dF_{X|Y}(x|y)\right]dF_Y(y) \tag{7}$$

Substituting for $E\{\varphi_i(X)|Y = y\}$, we have

$$\rho_{ij} = \int_{-\infty}^{\infty} \theta_j(y) \sum_{\ell=0}^{\infty} k_{i\ell}\, \theta_\ell(y)\, dF_Y(y) \tag{8}$$

On interchanging the integration and summation and using Eq. (4), we obtain $\rho_{ij} = k_{ij}$, $\forall j \geq 0$ and sufficiency is proved by a converse argument.▮

A similar remark can be made by interchanging $\varphi_i(x)$ and $\theta_j(y)$ and interchanging rows and columns. Note that the rows and columns of $[\rho_{ij}]$ are determined by the conditions given in Remark 2. Also it is clear that either the equation

$$E\{\varphi_i(X)|Y = y\} = \rho_{ii}\,\theta_i(y) \qquad , \forall i \geq 0 \tag{9}$$

or the equation

$$E\{\theta_j(Y)|X = x\} = \rho_{jj}\,\varphi_j(x) \qquad , \forall j \geq 0 \tag{10}$$

constitutes a set of necessary and sufficient conditions for $[\rho_{ij}]$ to be diagonal.

The concept of the diagonalness of the coefficient matrix $[\rho_{ij}]$ can be generalized by considering matrices which are zero except for the main diagonal and for the first i diagonals above and the first j diagonals below the main diagonal.

Definition 1. If a bivariate density f(x,y) is expressible as in Eq. (1) where $\{\varphi_m(x)\}$ and $\{\theta_n(y)\}$ are orthonormal polynomials, then f(x,y) is in the *class* Γ_{ij} of bivariate densities if and only if $\rho_{mn} = 0$ for $n-m>i$ and for $m-n>j$ where $i,j=0,1,2,\ldots$.

Note that the class Γ_{00} is the same as the class Λ of Barrett and Lampard [2]. The class Γ_{ij} can be characterized by the conditional expectations. The following theorem gives this characterization. The special case where i=0 and j=0 is due to Brown [3].

Theorem 1. If the bivariate density f(x,y) is expressible as in Eq. (1) where $\varphi_m(x)$ and $\theta_n(y)$ are orthonormal polynomials, of order m and n respectively, then f(x,y) is in the class Γ_{ij} if and only if

(i) $E\{X^k|Y = y\}$ is a polynomial of order $\leq k+i, \forall k \geq 0$,

and

(ii) $E\{Y^k|X = x\}$ is a polynomial of order $\leq k+j, \forall k \geq 0$;

for $i,j = 0,1,2,\ldots$.

Proof: Assume f(x,y) is in class Γ_{ij}. Then the conditional expectation $E\{X^k|Y = y\}$ is

$$E\{X^k|Y = y\} = \int_{-\infty}^{\infty} x^k\, f_{X,Y}(x,y)\,dx \tag{11}$$

On substituting for $f_{X,Y}$ from Eq. (1) and interchanging the order of integration and summation, we have

$$E\{X^k|Y = y\} = \sum_{m=0}^{\infty} \sum_{\substack{n=m-j \\ n\geq 0}}^{m+i} \rho_{mn}\ \theta_n(y) \int_{-\infty}^{\infty} \varphi_m(x)\ x^k\ f_X(x)dx \qquad (12)$$

Since x^k is representable as a linear combination of $\varphi_0(x)$, $\varphi_1(x),\ldots,\varphi_k(x)$, the integral in Eq. (12) is zero for $m>k$; therefore $E\{X^k|Y^k= y\}$ is a polynomial of order less than or equal to $k+i$, $\forall k \geq 0$. Similarly, $E\{Y^k|X = x\}$ is a polynomial of order less than or equal to $k+j$, $\forall k \geq 0$. This implies that $E\{\varphi_m(X)|Y=y\}$ is a polynomial of order less than or equal to $m+i$; therefore, by Eq. (7), $\rho_{mn} = 0$ for $n > m+i$. In the same way, if $E\{Y^k|X=x\}$ is a polynomial of order less than or equal to $k+j$, $\forall k\geq 0$, then $\rho_{mn} = 0$ for $m-n>j$. ∎

It should be clear that the rows and columns of $[\rho_{ij}]$ are determined through Remarks 1 and 2 while the number of diagonals having non-zero entries above and below the main diagonal can be controlled independently through Theorem 1. Most of the observations and conditions on the forms of the coefficient matrix given so far are also true when $\{\varphi_i(x)\}$ and $\{\theta_i(y)\}$ are not polynomials, Remark 1 is not true, since $\varphi_0(x)$ and $\theta_0(y)$ may not be unity. Remark 2 is true without modification.

III. CONDITIONS FOR THE EXPANSIONS OF A CLASS OF BIVARIATE DISTRIBUTIONS TO BE DIAGONAL

In this section, we investigate the case where X and Y are related by Eq. (2) and X and N are independent. As previously mentioned, this is a common restriction and the class Λ_o under investigation is a significant set of bivariate distributions. It is interesting to note that the restriction given by Eq. (2), when taken by itself, implies that the coefficient matrix $[\rho_{ij}]$ of the polynomial expansion is triangular.

Theorem 2. If a bivariate density has an expansion as in Eq. (1) with orthonormal polynomials and if Eq. (2) holds with X and N independent, then the coefficient matrix $[\rho_{ij}]$ is triangular.

Proof: For all $j\geq 0$, if we let

$$\theta_j(y) = \sum_{\ell=0}^{j} b_{j\ell}\ y^{\ell} \qquad (13)$$

where the b_{ij} are constant coefficients, then

$$E\{\theta_j(Y)|X = x\} = E\{ \sum_{\ell=0}^{j} b_{j\ell}\ Y^{\ell}|X = x\} \qquad (14)$$

On substituting for Y from Eq. (2) and interchanging the summations with expectation, we obtain

$$E\{\theta_j(Y)|X = x\} = \sum_{k=0}^{j} x^k \sum_{\ell=k}^{j} b_{j\ell} \binom{\ell}{k} E\{N^{\ell-k}\} \qquad (15)$$

This equation is true for all $j \geq 0$; therefore, by Theorem 1, all entries below the main diagonal of the coefficient matrix are zero.∎

It follows from Eq. (15) that $E\{\theta_j(Y) | X = x\}$ is a polynomial of order j; therefore by Remark 2, we have, for all $j \geq 0$,

$$E\{\theta_j(Y) | X = x\} = \sum_{\ell=0}^{j} \rho_{\ell j} \, \varphi_\ell(x) \tag{16}$$

If we let,

$$\varphi_\ell(x) = \sum_{k=0}^{\ell} a_{\ell k} \, x^k \tag{17}$$

where the a_{ij} are constant coefficients, and interchange the summations, Eq. (16) becomes

$$E\{\theta_j(Y) | X = x\} = \sum_{k=0}^{j} x^k \sum_{\ell=k}^{j} \rho_{\ell j} \, a_{\ell k} \tag{18}$$

It is clear from Eqs. (15) and (18) that

$$\sum_{\ell=k}^{j} b_{j\ell} \binom{\ell}{k} E\{N^{\ell-k}\} = \sum_{\ell=k}^{j} \rho_{\ell j} \, a_{\ell k} \tag{19}$$

for $0 \leq k \leq j$ and for all $j \geq 0$.

Now let us define the matrices $B = [b_{ij}]$, $A = [a_{ij}]$, $\rho = [\rho_{ij}]$ and $L = [L_{ij}] = \left[\binom{i}{j} \; E \, N^{i-j}\right]$ and the vectors

$$\underline{X}^T = \left[1 \; X \; X^2 \ldots\right] \quad , \quad \underline{\varphi(X)}^T = \left[\varphi_0(X) \; \varphi_1(X) \ldots\right],$$

and

$$\underline{Y}^T = \left[1 \; Y \; Y^2 \ldots\right] \quad , \quad \underline{\theta(Y)}^T = \left[\theta_0(Y) \; \theta_1(Y) \ldots\right]$$

where T denotes transpose. Then Eqs. (17), (13), and (19) can be expressed as follows, keeping in mind that B, A and L are triangular, with all entries zero above the main diagonal:

$$\underline{\varphi(x)} = A \, \underline{x} \quad , \quad \underline{\theta(y)} = B \, \underline{y} \tag{20}$$

and

$$B \, L = \rho^T A \tag{21}$$

On rearranging Eq. (21), it is clear that, for the class of bivariate distributions satisfying Eq. (2) with X and N independent, the coefficient matrix of the expansion is determined explicitly by

$$\rho^T = B \, L \, A^{-1} \tag{22}$$

when A^{-1} exists. The matrix A and its inverse exist for all orders, if the distribution function of X has infinitely many points of increase. The matrix A and its inverse exist for order up to m, if the distribution function has m points of increase. In passing, let us note that

Remark 3. The matrix A^{-1} is a triangular matrix, such that

$$A^{-1} A^{-1^T} = \overline{X} \tag{23}$$

where $\overline{X} = \left[\overline{X}_{ij}\right] = \left[E\ X^{i+j}\right]$ is the Hankel matrix of the r.v.X.

Proof. On multiplying $\underline{\varphi(x)} = A\ \underline{x}$ by its transpose and taking the expectation, we have

$$E\{\underline{\varphi(X)}\ \underline{\varphi(X)}^T\} = A\ E\{\underline{X}\ \underline{X}^T\}\ A^T \tag{24}$$

Since the $\{\varphi_i(x)\}$ are orthonormal and $E\{\underline{X}\ \underline{X}^T\} = \overline{X}$, Eq. (24) becomes

$$I = A\ \overline{X}\ A^T \tag{25}$$

where I is the identity matrix. On rearranging Eq. (25) we obtain Eq. (23).∎

Note that a necessary and sufficient condition for the factorization of $\overline{X}$ in the manner of Eq. (23) is that $\overline{X}$ be positive definite, which, in the context of the Hankel matrices of random variables, means that the distribution function must have infinitely many points of increase. This is the same condition required for the existence of A and A^{-1}. We would point out that there is a simple but laborious recursive procedure to determine A^{-1} from $\overline{X}$. We have

$$A^{-1}_{ij} = \frac{1}{A^{-1}_{jj}} \left(E\ X^{i+j} - \sum_{k=0}^{j-1} A^{-1}_{ik}\ A^{-1}_{jk}\right),\quad i \geq j \tag{26}$$

from Eq. (26), the entries of A^{-1} can be determined.

Conditions for the matrix ρ to be diagonal are determined from Eq. (22) by expanding that equation and requiring

$$\rho_{ji} = \rho_{jj}\ \delta_{ij} = \sum_{\ell=j}^{i} \sum_{m=j}^{\ell} b_{i\ell} \binom{\ell}{m}\ E\ N^{\ell-m}\ A^{-1}_{mj},\ \forall i,j \geq 0 \tag{27}$$

After rearranging, Eq. (27) is seen to be equivalent to

$$B^{-1}_{\ell j} = \frac{1}{\rho_{jj}} \sum_{m=j}^{\ell} \binom{\ell}{m}\ E\ N^{\ell-m}\ A^{-1}_{mj}\ ,\quad \forall \ell, j \geq 0 \tag{28}$$

It follows from Eq. (19) that $\rho_{jj} = b_{jj}/a_{jj}$, and, since A and B are both triangular, $\rho_{jj} = A^{-1}_{jj}/B^{-1}_{jj}$. On substituting for ρ_{jj}, the necessary and sufficient conditions for ρ to be diagonal become

$$B^{-1}_{\ell j}/B^{-1}_{jj} = \sum_{m=j}^{\ell} \binom{\ell}{m} E\ N^{\ell-m} A^{-1}_{mj}/A^{-1}_{jj}\ ,\quad \forall \ell, j \geq 0 \tag{29}$$

The matrix B^{-1} can be obtained from the Hankel matrix $\overline{Y}$ of the random variable Y in the same manner as A^{-1} is obtained from $\overline{X}$.

It is seen from Eq. (26) and a corresponding equation for B^{-1} that Eq. (29) is true for j=0 for this class of bivariate distributions, corresponding to the fact that the first row of ρ is zero except for the first entry. Similarly, it follows from Eq. (27) that the k-th row of ρ being zero except for the diagonal entry is equivalent to Eq. (29) holding for j=k and $\ell \geq 0$. Furthermore, it follows from Eq. (27) that, for any k, $i_o \geq 0$, $\rho_{ki} = \delta_{ki}\, \rho_{kk}$, where $0 \leq i \leq i_o$, if and only if

$$B^{-1}_{\ell k}/B^{-1}_{kk} = \sum_{m=k}^{\ell} \binom{\ell}{m} E\, N^{\ell-m}\, A^{-1}_{mk}/A^{-1}_{kk}\,, \quad 0 \leq \ell \leq i_o \tag{30}$$

Despite the elegant formulation, the conditions given in Eq. (29) are not very transparent. However, the first few of these conditions can be transformed into a more meaningful expression. On using Eq. (26) and a corresponding equation for B^{-1}, Eq. (29) becomes, for j=1,

$$\frac{EY^{\ell+1} - EY\ EY^{\ell}}{\text{Var } Y} = \sum_{m=1}^{\ell} \binom{\ell}{m} E\, N^{\ell-m}\, \frac{EX^{m+1} - EX\ EX^{m}}{\text{Var } X} \tag{31}$$

Substituting for Y from Eq. (2) and rearranging Eq. (31), we obtain

$$\sum_{m=1}^{\ell} \binom{\ell}{m} E\, N^{\ell-m}\, \frac{EX^{m+1} - EX\ EX^{m}}{\text{Var } X} = \sum_{m=1}^{\ell} \binom{\ell}{m} EX^{\ell-m}\, \frac{EN^{m+1} - EN\ EN^{m}}{\text{Var } N} \tag{32}$$

for all $\ell \geq 0$. For $\ell=0$ and 1, Eq. (32) holds trivially. For $\ell=2$, Eq. (32) becomes

$$\frac{E[X-EX]^3}{\text{Var } X} = \frac{E[N-EN]^3}{\text{Var } N} \tag{33}$$

which is equivalent to $\rho_{12} = 0$, from Eq. (30). We can summarize the preceeding conditions and their interrelation schematically as follows:

$$\begin{tikzcd}
\text{Eq. (29)} \arrow[r, Rightarrow, "j=k"'] \arrow[d, Leftrightarrow] & \text{Eq. (29) for } j=k \arrow[r, Rightarrow, "k=1"'] \arrow[d, Leftrightarrow] & \text{Eq. (32)} \arrow[r, Rightarrow] \arrow[d, Leftrightarrow] & \text{Eq. (33)} \\
\rho_{ji} = 0 \text{ for } j \neq i,\ i \geq 0 \arrow[r, Rightarrow, "j=k"] & \rho_{ki} = 0 \text{ for fixed } k,\ i \neq k \arrow[r, Rightarrow, "k=1"] & \rho_{1i} = 0,\ i \neq 1 &
\end{tikzcd} \tag{34}$$

As shown in the diagram of Eq. (34), Eq. (32) constitutes a necessary set of conditions for ρ to be diagonal. If either N or X is specified through its moments, the moments of the other are determined in terms of its first two moments by successive application of Eq. (32). However, some problems may arise. First, with N specified, it is not clear whether an arbitrary choice of the first two moments of X will result in a sequence which are legitimate moments. However, there is always some EX and EX^2 which will lead to legitimate moments; for example, EX=EN and $EX^2=EN^2$. This is the case when all the moments of X are identical to all the moments of N. Another problem might be that the moments of X may not determine uniquely the distribution function of X. This last difficulty can sometimes be eliminated by checking whether the moments satisfy sufficient conditions to determine uniquely the distribution function (Shohat-Tamarkin [7], p. 19). Finally, for the N or X specified and the

X or N obtained, we must check that the expansion is in fact diagonal. Using Eq. (32), we obtain the following result:

Theorem 3. Assume that the bivariate density of X and Y is expressible as in Eq. (1), where Y is given by Eq. (2), and $\{\varphi_m(x)\}$ and $\{\theta_n(y)\}$ are orthonormal polynomials. For N Gaussian, Gamma, or Poisson distributed, Eq. (32) is true, if and only if, X also is Gaussian, Gamma, or Poisson distributed, respectively, except possibly for a translation and scaling depending on the choice of EX and EX^2. In all three cases the moments of X determine uniquely its distribution function and the expansion is diagonal.

Proof. The proofs for these cases are relatively long although straightforward. Lack of space precludes their inclusion here although they can be found in [8]. $\blacksquare$

We repeat that Eq. (32) is only a necessary condition for diagonal ρ. For example, for N specified to be uniformly or double-exponentially distributed, Eq. (32) implies that X is uniformly or double-exponentially distributed, respectively. But in both cases the expansion fails to be diagonal. Or, for any N and X identically distributed, Eq. (32) is satisfied; however, on using Eq. (29) for j=2, it can be shown that the expansion is not diagonal in general. Moreover, N and X being identically distributed and symmetric with respect to their means does not insure a diagonal ρ.

As in the j=1 case, for j>1, Eq. (29) can be expressed in terms of the moments of X and N. But, even for j=2, the expressions become prohibitively complicated.

Note that the class Λ_o of bivariate distributions have been characterized [8] from a different point of view. In [8], it is shown that the N and X which makes the second and third rows of ρ zero except for the diagonal element also makes the other rows of ρ zero except for the diagonal elements. In other words, Eq. (29) for j=1 and 2 constitutes a set of necessary and sufficient conditions for ρ to be diagonal.

From still another point of view, Balakrishnan ([9], Thm. 3) gave conditions on the characteristic functions of N and X so that the least-mean-square estimate of X is a linear function of Y. This is equivalent to the second row of ρ being zero except for the diagonal element, which is in turn equivalent to Eq. (32). In other words, Eq. (32) is also a set of necessary and sufficient conditions for the least-mean-square estimate of X to be linear in the observation Y. To clarify the interrelation of these various conditions, we give a diagram as shown in Fig. 1 on the next page. Here $P_m(\cdot)$ is a polynomial of order m, $\phi_X(t)$ and $\phi_N(t)$ are the characteristic functions of X and N, and $\beta = p_{10}/(1-p_{11})$ and $\alpha = p_{11}/(1-p_{11})$ where $p_1(\cdot) = p_{10} + p_{11}(\cdot)$. Note that, if the relationship in any rectangular box is true for all k≥0, then this relationship is equivalent to ρ being diagonal.

For the sake of completeness, we give another set of necessary conditions for ρ to be diagonal. We already know from Theorems 1 and 2 that a necessary and sufficient set of condi-

$$\begin{tikzcd}
\boxed{E\{\varphi_k(X)\,|\,Y=y\} = \rho_{kk}\,\theta_k(y)} \arrow[r, Leftrightarrow] \arrow[d, Rightarrow, shift left, "k=1"] & \boxed{\rho_{ki} = 0 \text{ for all } i \neq k} \arrow[d, Leftrightarrow] \\
\boxed{\begin{array}{c} E\{X^m\,|\,Y=y\} = P_m(y) \\ 1 \le m \le k \end{array}} \arrow[u, Rightarrow, shift left] \arrow[d, Rightarrow, shift right] & \boxed{\begin{array}{c} \dfrac{B^{-1}_{\ell k}}{B^{-1}_{kk}} = \sum\limits_{m=k}^{\ell} \binom{\ell}{m} E\,N^{\ell-m} \dfrac{A^{-1}_{mk}}{A^{-1}_{kk}} \\ \forall \ell \ge 0 \end{array}} \\
\boxed{\emptyset_X(t) = e^{\beta i t}\,\emptyset_N^{\alpha}(t)} \arrow[u, Rightarrow, shift right, "k=1"'] &
\end{tikzcd}$$

Fig. 1 - Interrelation of Conditions

tions is that $E\{X^k|Y = y\}$ be a polynomial of order k, $\forall k \ge 0$. Another set of necessary conditions is given by

Theorem 4. For Y given by Eq. (2), for all moments of X and N existing and for any k, if

$$E\{X^k|Y = y\} = \sum_{i=0}^{k} p_{ki}\, y^i \qquad (35)$$

then

$$\sum_{j=0}^{m} \binom{m}{j} EN^{m-j} EX^{k+j} = \sum_{i=0}^{k} p_{ki} \sum_{j=0}^{m+i} \binom{m+i}{j} EN^{m+i-j} EX^j \qquad (36)$$

for all $m \ge 0$.

Proof. Since $E|X^k| < \infty$ and $E|X^k Y^m| < \infty$ by definition, it follows that ([10], p. 22)

$$E\{X^k Y^m\} = E\left\{Y^m E\{X^k|Y\}\right\} \quad , \quad \forall m \ge 0 \qquad (37)$$

Substituting for $E\{X^k|Y\}$ from Eq. (35), we have

$$E\{X^k Y^m\} = \sum_{i=0}^{k} p_{ki}\, EY^{m+i} \quad , \quad \forall m \ge 0 \qquad (38)$$

Now, on substituting for Y from Eq. (2), we obtain Eq. (36). ▮

Note that the converse of this theorem is not necessarily true because $E\left\{Y^m E\{X^k|Y\}\right\} = E\left\{Y^m \sum_{i=0}^{k} p_{ki}\, Y^i\right\}$, for all positive integer values of m, might not imply that $E\{X^k|Y\} = \sum_{i=0}^{k} p_{ki} Y^i$.

Equation (36) for $0 \le m \le k$ can be solved for p_{ki}, $i=0,\ldots,k$. On substituting these p_{ki} back into Eq. (36) for all $k \ge 0$, the result represents a necessary set of conditions on the moments of X and N for diagonal ρ. Note that, although Eq. (29) can also be expressed in terms of the moments, Eq. (36) is more transparent as far as the conditions on moments are concerned. For k=0, Eq. (36) is trivially true. For k=1, on determining p_{10} and p_{11} and substituting them back into Eq. (36) and rearranging, it becomes clear that Eq. (36) is identical to Eq. (32).

ACKNOWLEDGEMENTS

This research is supported by the National Science Foundation under Grant GK-24187 and by the Air Force Office of Scientific Research under Grant AFOSR-71-2101.

REFERENCES

1. J.B. Thomas, An Introduction to Statistical Communication Theory, Chapter 6, New York: Wiley, 1969.

2. J.F. Barrett and D.G. Lampard, "An Expansion for Some Second-Order Probability Distribution in Orthogonal Polynomials", IRE Trans. Information Theory, Vol. IT-1, pp. 10-15, March 1955.

3. J.L. Brown, "A Criterion for the Diagonal Expansion of a Second-Order Probability Distribution in Orthogonal Polynomials", IRE Trans. Information Theory, Vol. IT-4, p. 172, December 1958.

4. E. Wong and J.B. Thomas, "On Polynomial Expansions of Second- Order Distributions", SIAM J. Appl. Math., Vol. 10, pp. 507-516, September 1962.

5. N.I. Akhiezer and I.M. Glazman, Theory of Linear Operators in Hilbert Space, Vol. I, New York: Ungar, 1961.

6. G. Sansone, Orthogonal Functions, New York: Interscience, 1959.

7. J.A. Shohat and J.D. Tamarkin, The Problem of Moments, Am. Math. Soc., New York, 1943.

8. H. Derin, "On the Properties of Series Expansions of Bivariate Distributions", Doctoral Dissertation, Department of Electrical Engineering, Princeton University, January 1972.

9. A.V. Balakrishnan, "On a Characterization of Processes for which Optimal Mean-Square Systems are of Specified Form", IRE Trans. Information Theory, Vol. IT-6, pp. 490-500, September 1960.

10. J.L. Doob, Stochastic Processes, New York: Wiley, 1953.

REDUCED MEMORY INFORMATION PROCESSORS FOR POINT PROCESSES*

IZHAK RUBIN
System Science Department
School of Engineering and Applied Science, University of California
Los Angeles, California 90024

ABSTRACT

The problems of reduced-memory modeling and processing of regular point processes are studied. m-memory processes and processors are defined as those whose present (incremental) behavior depends only on the present observation of counts, and the stored values of the preceding m instants of occurrence.

Employing a Bayes optimization criterion, the best m-memory processor is derived. Characterization theorems for m-memory processes are obtained. A 0-memory process is shown to be a Markov counting process, while a process is 1-memory iff its associated occurrence sequence is Markov. Homogeneous 1-memory processes are identified as renewal processes. These results are then applied to study reduced-memory detection schemes for Doubly-stochastic Poisson Processes.

1. INTRODUCTION

We consider an honest counting process $\{N(t),\ 0 \le t \le T\}$, whose state space is the nonnegative integers and which is defined on a probability space (Ω_N, β_N, P_N). $N(t)$ denotes the number of point occurrences in $[0,t)$, $N(0) = 0$. We denote by $\mathscr{B}_t$ the Borel field generated by $\{N(\tau),\ 0 \le \tau \le t\}$. We assume that this process is a r.p.p. (regular point process) [1]; i.e., the following limits hold $\forall t \varepsilon [0,T]$

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} [1 - P\{N(t+\Delta t) = N(t) | \mathscr{B}_t\}] = \lim_{\Delta t \to 0} \frac{1}{\Delta t} [P\{N(t+\Delta t) = N(t) + 1 | \mathscr{B}_t\}]$$

$$\stackrel{\Delta}{=} \lambda_{N(t)}(t, \mathscr{B}_t) \equiv \lambda(t,\omega) \quad , \tag{1a}$$

and

$$E\{\lambda_{N(t)}(t, \mathscr{B}_t)\} < \infty \quad . \tag{1b}$$

Furthermore, letting (Ω_1, β_1, P_1) denote the sample-functions space of $\{N(t),\ 0 \ge t \ge T\}$, for each realization of the point process over $[0,T]$, $\omega \varepsilon \Omega_1$, a realization of $\lambda(t,\omega)$ (called the intensity process) is assumed to be a positive bounded piecewise-continuous function over $[0,T]$, and is taken to be left-continuous. The sample functions of the r.p.p. have only jump discontinuities and we choose the sample functions to be left continuous. Associated with the counting process $\{N(t);\ 0 \le t \le T\}$ is the sequence of point occurrences $\{W_n;\ n=0, 1,2,\dots\}$, $W_o = 0$, where W_n denotes the instant of the n^{th} occurrence. Consequently,

$$W_{N(t)}(\omega) = \inf_{\tau}\{\tau \ge 0:\ N(\tau,\omega) = N(t,\omega)\} \tag{2}$$

denotes the instant of occurrence preceding t, defined over $[0,t)$.

---

*This work was supported by the office of Naval Research, under Grant No. N00014-69-H-0200-4041.

Since a r.p.p. is characterized by $\lambda(t,\omega)$, the intensity process contains the information transferred by the process. This is represented as follows. Let $\{S(t),\ 0 \le t \le T\}$ denote the information-bearing process. Due to noise interferences or uncertainties concerning its nature, one has to consider $S(t)$ as a stochastic process, defined on a probability space $(\Omega_S,\ \beta_S,\ P_S)$. This message process is now assumed to causally modulate the intensity function of the point process. Thus, if S_t denotes a realization $\{S(\tau),\ 0 \le \tau \le t\}$ the counting process possesses the intensity function $\lambda_{N(t)}(t, \mathscr{B}_t,\ S_t) \equiv \lambda(t,\omega,S_t)$, $\forall S_t$ and $\omega \varepsilon \Omega_1$. It is shown in [1] that the observed compound point process $\{\tilde{N}(t),\ 0 \le t \le T\}$ is again a r.p.p. whose intensity function is given by

$$\tilde{\lambda}_{N(t)}(t,\mathscr{B}_t) = E\{\lambda_{N(t)}(t,\mathscr{B}_t,S_t)\,|\mathscr{B}_t\} \triangleq \tilde{\lambda}_{N(t)}(t,\mathscr{B}_t), \tag{3}$$

provided $S(t)$ is smooth so that $\lambda_{N(t)}(t,\ \mathscr{B}_t, S_t)$ has the properties of an intensity process.

Detection, filtering and estimation operations for the r.p.p.'s in the system, require calculation of the a posteriori densities. Specifically, consider a binary detection problem, where $\{S^{(j)}(t),\ 0 \le t \le T\}$ is assumed to be the message-process under hypothesis H_j, $j = 1,2$. The log-likelihood-ratio processor is then shown in [1] to be given explicitly by

$$\begin{aligned} \ell n\ \Lambda_T(\omega) = & \sum_{i=1}^{N(T)} \ell n\left[\hat{\lambda}^{(2)}_{i-1}\ (W_i,\ldots,W_1) - \hat{\lambda}^{(1)}_{i-1}\ (W_i,\ldots,W_1)\right] \\ & - \sum_{i=1}^{N(T)} \int_{W_{i-1}}^{W_i} \left[\hat{\lambda}^{(2)}_{i-1}\ (u,\ W_{i-1},\ldots,W_1) - \hat{\lambda}^{(1)}_{i-1}(u,\ W_{i-1},\ldots,W_1)\right] du \\ & - \int_{W_{N(T)}}^{T} \left[\hat{\lambda}^{(2)}_{N(T)}(u,\ W_{N(T)},\ldots,W_1) - \hat{\lambda}^{(1)}_{N(T)}(u,\ W_{N(T)},\ldots,W_1)\right] du \end{aligned} \tag{4}$$

where

$$\hat{\lambda}^{(j)}_{N(t)}(t,\mathscr{B}_t) = E\{\lambda_{N(t)}(t,\ \mathscr{B}_t, S_t)|\mathscr{B}_t, H_j\}\ ,\ j = 1,2, \tag{5}$$

and

$$\lambda_{N(t)}(t,\ W_{N(t)},\ldots,W_1) \equiv \lambda_{N(t)}(t,\ \mathscr{B}_t)(\text{defining } \lambda_{i-1}\left(u,\ W_{i-1},\ldots,W_1\right) \triangleq \lambda_o(u)\Big).$$

A similar corresponding expression is derived in [1] for the posterior density of any r.p.p.

Thus, to extract information from the incoming r.p.p., one has to be able to calculate at each instant t its intensity function $\lambda_{N(t)}(t,W_{N(t)},\ldots,W_1)$. It is then used for likelihood processing as shown by (4). One must therefore have a complete statistical characterization of $\lambda(t,\omega)$ (or $\hat{\lambda}(t,\omega)$ for compound r.p.p.'s). Equivalently, at each instant all past point occurrences must be incorporated to generate the present intensity function. However, this situation is practically infeasible due to the following main factors:

1) Modeling: By definition, $\lambda_{N(t)}(t,\mathscr{B}_t)\,\Delta t$ is asymptotically $(\Delta t \downarrow 0)$ the conditional probability of an occurrence in $[t,\ t+\Delta t)$, given the evolution of the counting process over $[0,t)$. This interpretation indicates that the statistical character of $\lambda(t,\omega)$ could be established, by using learning samples or physical considerations concerning the counting stochastic mechanism. However, these operations would be practically feasible only if the above function is assumed to depend on at most a few of the preceding instants of occurrence.

2) Processor complexity: It follows from Eq. (4) that the processor must store and process at each instant t all the past observations, i.e., $\{N(t), W_{N(t)} \dots, W_1\}$.

3) Estimator complexity: When the message signal S(t) is assumed to be a stochastic process, one has to employ a least-squares estimator (5) to derive the intensity function. Since the latter is $\mathscr{B}_t$ measurable, the estimator will require generally, at each instant, all past observations.

All these factors indicate the necessity of reducing the effective memory utilized to calculate the instantaneous value of the intensity function. We also note that the observed regular point processes are generally nonstationary, and being usually associated with some finite memory physical system, the dependence of their present behavior $\{N(t)\}$ on the past $\{N(u), 0 \le u < t\}$, is reduced rapidly as u decreases. Assuming the latter to be the case, and that a simple sufficient statistic does not exist, as is generally the case, we conclude that it is practically desirable to process, at each instant, only a reduced number of the recent occurrence observations. In addition, as Eq. (6) indicates, one should generally utilize an instantaneous count observation, and we will thus assume throughout the paper that a counter is available. The memory complexity is therefore directly associated with the number of recent event occurrences stored at each instant of time. This number will then be used to indicate the memory of the processor.

In this paper we resolve the above complexity problems by deriving the best reduced-memory r.p.p.'s to approximate the given r.p.p.'s. The resulting (likelihood) processors, as well as the intensity estimators, will then also have a reduced-memory character, and subsequently be structurally much less complex.

2. REDUCED-MEMORY PROCESSORS AND r.p.p.'s

Definitions

We consider a compound r.p.p. $\{N(t), 0 \le t \le T\}$ whose intensity function is $\hat{\lambda}_{N(t)}(t, B_t)$. Define now the Borel field generated by the m occurrences preceding t as

$$\mathscr{B}_t^m \equiv \sigma\{N(t), W_{N(t)}, W_{N(t)-1}, \dots, W_{N(t)-m+1}\} \; ; \; m=1,2,\dots$$

$$\mathscr{B}_t^0 \equiv \sigma\{N(t)\} \; , \tag{6}$$

where $W_i = 0$ for $i \le 0$. From the definition it is clear that

$$\mathscr{B}_t^m \subset \mathscr{B}_t^n \subset \mathscr{B}_t \quad \forall n > m, \text{ and } \mathscr{B}_t^\infty \equiv \bigcup_n \mathscr{B}_t^n = \mathscr{B}_t \; . \tag{7}$$

The intensity function $\hat{\lambda}_{N(t)}(t, \mathscr{B}_t)$ is $\mathscr{B}_t$ measurable by its definition (7), but not necessarily $\mathscr{B}_t^m$ measurable. Hence, we introduce the following definitions.

Definition 1: A random function measurable w.r.t. B_t will be called an m-memory random function if it is also measurable w.r.t. $\mathscr{B}_t^m$, $\forall t \in [0,T]$, $m=0,1,2,\dots$

Definition 2: A processing scheme (for point processes) will be said to be an m-memory processor over [0,T], if it utilizes at each $t \in [0,T]$ only m-memory random functions.

Definition 3: A point process will be said to be an m-memory point process over [0,T], if it is a r.p.p. whose intensity function is an m-memory function.

Thus, an m-memory function will utilize at t only observations of $\{N(t), W_{N(t)},\ldots,W_{N(t)-m+1}\}$ if $m \geq 1$, and $\{N(t)\}$ alone if $m = 0$. For likelihood processing schemes of r.p.p.'s, one readily deduces from Eq. (4):

Lemma 1: An optimal likelihood processor of r.p.p.'s is m-memory $(m \geq 1)$ if and only if the incoming processes are m-memory; i.e., their intensity functions $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$ are m-memory functions.

The above definition 2 and Lemma 1 form thus the basis for the derivation of an m-memory processor.

m-Memory Approximating Processes and the Best m-Memory Processor

Assume the present processes are compound r.p.p.'s, and consider a two hypotheses detection problem as in Section 1. Lemma 1, indicates that for the optimal likelihood processor (4) to be m-memory, the intensity functions $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$ must be $\mathcal{B}_t^m$ measurable. Consequently, the incoming r.p.p.'s are required to be m-memory. Since this will not usually be the case, the optimal m-memory processor is sought. Assuming costs C_{ij}, i, j = 1,2, C_{ij} being the cost of deciding in favor of H_i given that H_j is the true state, we use a *Bayes* optimization criterion. In particular, taking $C_{ij} = 1-\delta_{ij}$, $i,j = 1,2$, the desired processor is that which *minimizes the probability of error*.

Before proceeding to obtain the Bayes solution, we present the following approach. To obtain an m-memory processor we should approximate the present r.p.p.'s by m-memory r.p.p.'s. As a r.p.p. is characterized by its intensity function, we have to use the best, in the appropriate sense, $\mathcal{B}_t^m$ measurable approximation of $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$. The resulting intensity will characterize then the corresponding approximating m-memory r.p.p. model and processor. Assume now that one chooses as an index of fitness of the desired approximation the least-squares error (MMSE) criterion. Denote the m-memory approximating intensity by $\mu_{N(t)}(t,\mathcal{B}_t^m)$. The (instantaneous) m.s. error is

$$e(t) = E\left\{[\hat{\lambda}_{N(t)}(t,\mathcal{B}_t) - \mu_{N(t)}(t,\mathcal{B}_t^m)]^2\right\}. \tag{8}$$

The following is well known.

Lemma 2: The least-squares m-memory estimate of $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$ is given by

$$\hat{\mu}_{N(t)}(t,\mathcal{B}_t^m) = E\{\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)\,|\,\mathcal{B}_t^m\}. \tag{9}$$

Proof: One can write (10) as $e(t) = E\left\{E\{[\hat{\lambda}_{N(t)}(t,\mathcal{B}_t) - \mu_{N(t)}(t,\mathcal{B}_t^m)]^2\,|\,\mathcal{B}_t^m\}\right\}$. Minimizing the inner expectation term, we obtain (11), since $\mu_{N(t)}(t,\mathcal{B}_t^m)$ is $\mathcal{B}_t^m$ measurable. Q.E.D.

The resulting m.s. error is the expected conditional variance,

$$e_m(t) = E\{E[\hat{\lambda}^{(2)}_{N(t)}(t,\mathcal{B}_t)\,|\,\mathcal{B}_t^m] - E[\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)\,|\,\mathcal{B}_t^m]^2\}. \tag{10}$$

Since $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$ is a MMSE estimate of the intensity $\lambda_{N(t)}(t,\mathcal{B}_t,S_t)$ (see (3)), the following result is readily obtained, for any compound r.p.p..

Proposition 1: The least-squares m-memory estimate of $\hat{\lambda}_{N(t)}(t,\mathcal{B}_t)$ is given by

$$\hat{\mu}_{N(t)}(t,\mathcal{B}_t^m) = E\{\lambda_{N(t)}(t,\mathcal{B}_t,S_t)\,|\,\mathcal{B}_t^m\}. \tag{11}$$

Proof: Using relation (3), Lemma 2 and the iteration rule for conditional probabilities for $\mathscr{B}_t^m \subset \mathscr{B}_t$, one concludes

$$\begin{aligned}\hat{\mu}_{N(t)}(t,\mathscr{B}_t^m) &= E\{\hat{\lambda}_{N(t)}(t,\mathscr{B}_t)\,|\,\mathscr{B}_t^m\} = E\{E[\lambda_{N(t)}(t,\mathscr{B}_t,S_t)\,|\,\mathscr{B}_t]\,|\,\mathscr{B}_t^m\}\\ &= E\{\lambda_{N(t)}(t,\mathscr{B}_t,S_t)\,|\,\mathscr{B}_t^m\}\ . \qquad \text{Q.E.D.}\end{aligned}$$

We can, however, much strengthen the significance of the MMSE instantaneous intensity estimate (11). We show here that the latter is actually also the appropriate intensity to be incorporated in the best m-memory detection procedure, assuming a Bayes optimization criterion for choosing the decision procedure. Thus, in particular, the approximating m-memory r.p.p. whose intensity is given by (11) is the best, in the sense that its related processing (4) when associated with the threshold $\ell n(\pi_1/\pi_2)$ (where π_i is the prior probability of H_i, i=1,2), constitutes an m-memory processor which minimizes the error probability.

Theorem 1: Assume the observed process under H_i, to be a compound r.p.p. with the conditional intensity function $\lambda_{N(t)}^{(i)}(t,\mathscr{B}_t, S_t^{(i)})$, i=1,2. Assuming a sample function to be observed over [0,T] and a Bayes optimization criterion, the best m-memory detection procedure is given by (4) where $\hat{\lambda}_i^{(j)}(t, W_i,\ldots,W_1)$ is to be replaced by $\hat{\lambda}_i^{(j)}(t, W_i, W_{i-1},\ldots,W_{i-m+1})$ of (11); i.e., the best m-memory intensity is $\hat{\lambda}_i^{(j)}(t, W_i,\ldots,W_{i-m+1}) = E\{\lambda_{N(t)}^{(j)}(t,\mathscr{B}_t, S_t^{(j)})\,|\,N(t) = i, W_i,\ldots,W_{i-m+1}\}$ for $m \geq 1$, j=1,2, and for m=0: $\hat{\lambda}_i^{(j)}(t) = E\{\lambda_{N(t)}^{(j)}(t,\mathscr{B}_t, S_t^{(j)})\,|N(t) = i\}$, j=1,2.

Proof: For an m-memory processor, at each instant $t\varepsilon[0,T]$, we observe only $\{N(t), W_{N(t)},\ldots,W_{N(t)-m+1}\}$ when $m \geq 1$, and only $\{N(t)\}$ if m=0. Assume the rest of the past counts observations to constitute the realization of a stochastic process $\{\theta_t^{(j)}, t\varepsilon[0,T]\}$, when hypothesis H_j holds, j=1,2. Thus, assuming we are given the realization of $S_t^{(j)}$ and $\theta_t^{(j)}$, the intensity function of the observed point process, under H_j, is given by $\lambda_{N(t)}^{(j)}(t,\mathscr{B}_t^m, S_t^{(j)}, \theta_t^{(j)}) = \lambda_{N(t)}^{(j)}(t,\mathscr{B}_t, S_t^{(j)})$. Since we require the intensity to be, at each instant t, $\mathscr{B}_t^m$ measurable, we can regard $\{S_t^{(j)}\}$ and $\{\theta_t^{(j)}\}$ as "nuisance" stochastic processes. Under these conditions, it is well known that the Bayes solution is given by the generalized likelihood-ratio. The latter is generated as the ratio $\hat{f}_T^{(2)}(\omega)/\hat{f}_T^{(1)}(\omega)$, where $\hat{f}_T^{(j)}(\omega)$ is the joint occurrence density under H_j, compounded w.r.t. the distributions of $\{S_t^{(j)}\}$ and $\{\theta_t^{(j)}\}$. However, as defined in [1], the latter are the joint occurrence densities of the associated compound r.p.p.'s, for which Theorem 4 in [1] indicates that the intensity function is given by

$$\begin{aligned}\hat{\lambda}_{N(t)}^{(j)}(t,\mathscr{B}_t^m) \text{ where } \hat{\lambda}_{N(t)}^{(j)}(t,\mathscr{B}_t^m) &= E\{\lambda_{N(t)}^{(j)}(t,\mathscr{B}_t^m, S_t^{(j)}, \theta_t^{(j)})\,|\,\mathscr{B}_t^m, H_j\}\\ &= E\{\lambda_{N(t)}^{(j)}(t,\mathscr{B}_t, S_t^{(j)})\,|\,\mathscr{B}_t^m, H_j\}\ ,\end{aligned}$$

which proves our statement. Q.E.D.

Theorem 1 thus indicates that the best approximating m-memory intensity function at t, is obtained by averaging the given intensity function over S_t and over all possible preceding occurrences conditioned on the present count and the last m occurrences. The best approximating m-memory r.p.p. model then results. Its associated m-memory likelihood processor follows from (4).

3. m-MEMORY PROCESSES

0-memory Processes

Assume now that we wish to utilize, at each instant t, only the present observed count N(t) to calculate the intensity function (or, in the compound case, to causally estimate it). The processing to be performed and the appropriate 0-memory approximating model follow from Theorem 1. Thus, the best 0-memory likelihood-ratio processor is given by (4), when $\hat{\lambda}^{(j)}_{N(t)}(t, \mathscr{B}_t)$, j=1,2, is replaced by

$$\hat{\lambda}^{(j)}_{N(t)}(t) \triangleq \hat{\mu}^{(j)}_{N(t)}(t, \mathscr{B}^{o}_{t}) = E\{\lambda_{N(t)}(t, \mathscr{B}_t, S^{(j)}_{t}) | N(t), H_j\}. \qquad (12)$$

The problem that subsequently arises is to statistically characterize all 0-memory point processes. To solve this problem, we define a Markov point process as a r.p.p. whose counting process $\{N(t), 0 \leq t \leq T\}$ is Markov (or, equivalently, as a Pure-Birth process, [4], [5]). We can show the following property [2].

Theorem 2: A r.p.p. is 0-memory if and only if it is a Markov point process.

Thus, the likelihood processor obtained from (4) is 0-memory if and only if the incoming processes are Markov point processes. In other cases, the 0-memory processor obtained when (12) is incorporated, is the best 0-memory procedure. In particular, approximating the incoming r.p.p.'s by Markov point processes using (12), yields a 0-memory processor which minimizes the probability of error. From a modeling point of view, Theorem 2 incidates that if only 0-memory operations are desired, the appropriate approximating model for the incoming r.p.p., is that of a Markov point process. The resulting mean-square intensity error is given by (10).

1-memory Processes

Although the 0-memory processor requires only the measurement of the present count N(t) to calculate the intensity estimate (3), Eq. (4) indicates that for actual likelihood computation the instant of last occurrence, $W_{N(t)}$, need also be stored. Consequently, one may utilize this additional information for deriving a better intensity estimate; or, equivalently, a better approximating r.p.p. model. The latter is thus assumed now to be of a 1-memory nature. By Theorem 1, the best 1-memory approximating process will possess the intensity function

$$\hat{\mu}^{(i)}_{N(t)}(t, W_{N(t)}) = E\{\lambda_{N(t)}(t, \mathscr{B}_t, S^{(i)}_{t}) | N(t), W_{N(t)}, H_i\} \ . \qquad (13)$$

This intensity is incorporated in (4) to yield the best 1-memory (likelihood) processor.

As for the 0-memory case, it is important to characterize the class of 1-memory point processes. We can prove [2] the following theorem which provides this statistical characterization.

Theorem 3: A point process is 1-memory if and only if it is a r.p.p. whose associated random sequence of occurrence times, $\{W_n;\ n=1,2,\ldots\}$ is Markov.

From Theorem 3, we conclude that when a 1-memory processor is to be used the appropriate models of the incoming point processes are r.p.p.'s whose occurrence sequence is Markov. The transition density of the occurrence times, under H_i, is then seen to be given by

$$f^{(i)}(t, t_n) = \hat{\lambda}^{(i)}_{n}(t, t_n) \exp\left[- \int_{t_n}^{t} \hat{\lambda}^{(i)}_{n}(u, t_n) du \right] , \qquad (14)$$

where $W_0 = 0$ and $\hat{\lambda}_n^{(i)}(t, t_n)$ is given by (13). The 1-memory likelihood-ratio processor is so expressed as

$$\ln \Lambda_T = \sum_{i=1}^{N(T)} \ln \Lambda_i + \ln[\mathscr{F}^{(2)}(T,t_{N(T)})/\mathscr{F}^{(1)}(T,t_{N(T)})]$$

where

$$\Lambda_i = f^{(2)}(t_i, t_{i-1})/f^{(1)}(t_i, t_{i-1}) \tag{15}$$

for the realization $\{W_i = t_i;\ i=1,2,\ldots,N(T);\ 0 < t_1 < \ldots < t_{N(T)} \leq T\}$. Relation (14) follows by observing that

$$\lambda_n(t,t_n) \overset{\Delta}{=} \lim_{\Delta t \downarrow 0}(\Delta t)^{-1} P\{t \leq W_{n+1} < t+\Delta t | W_{n+1} \geq t,\ W_n = t_n\}$$

$$= f(t, t_n)/\mathscr{F}(t, t_n),$$

where $f(t_{n+1}, t_n)$ and $F(t_{n+1}, t_n)$ are the conditional density and distribution function, respectively, of W_{n+1} given W_n and $\mathscr{F}(\cdot) = 1-F(\cdot)$. Note that now one is required to provide (by using learning procedures, estimation and relation (14), or proper assumptions concerning the physical nature of the system) just the transition densities of the instants of occurrence of the incoming point process. Clearly, an m-memory point process is characterized as a r.p.p. whose occurrence sequence is m-order Markov.

Homogeneous 1-memory Processes

In practice, one can many times assume that a point process is time-homogeneous. An event occurrence at t will then depend on the previous occurrence, at $W_{N(t)}$, only through the epoch $Z(t) = t - W_{N(t)}$. That event may still depend also on the number of total past occurrences N(t). The 1-memory r.p.p.'s which possess these homogeneous properties, and subsequently constitute the best approximating 1-memory homogeneous processes, are characterized by the following theorems [2].

Theorem 4: A point process is 1-memory with an intensity function satisfying

$$\lambda_{N(t)}(t, W_{N(t)}) = \nu_{N(t)}(t - W_{N(t)}), \tag{16}$$

if and only if it is a r.p.p. whose associated random sequence of occurrences $\{W_n,\ n=1,2,\ldots\}$ is Markov with independent increments (i.e., with independent interarrival times).

For the time and state homogeneous case, we can prove the following:

Theorem 5: A point process is 1-memory with an intensity function satisfying

$$\lambda_{N(t)}(t, W_{N(t)}) = \nu(t - W_{N(t)}), \tag{17}$$

if and only if it is a renewal process.

Note: A renewal process is defined as a point process for which the intervals between occurrences T_i, i=1,2,... are independent and identically distributed [6].

Theorem 5 thus indicates that, under the homogeneity condition (17), the appropriate 1-memory statistical model is that of a renewal process. The interval density is then

$$f_{T_i}(x) = \nu(x) \exp\left[-\int_0^x \nu(u)du\right], \quad x \geq 0. \tag{18}$$

The best 1-memory likelihood-ratio processor is now given by (15), when $f(t,t_n) = f_{T_i}(t-t_n)$ is incorporated. This processor requires only the distribution of the intervals between occurrences, which must be estimated or assumed through physical considerations. The detector's performance characteristics are easily calculated from (15), since the terms these are now statistically independent [3].

4. APPLICATIONS-REDUCED MEMORY DETECTION OF DSPPs

Doubly-Stochastic Poisson Processes (DSPPs) serve as statistical models in photon communications [8] and various biological systems [7]. Given the realization of some stochastic process, to be called the message process, a DSPP is a Poisson process whose intensity function is specified in terms of the above realization. Assume the counting process $\{N(t), 0 \le t \le T\}$, $N(0) = 0$, is a doubly-stochastic Poisson process (DSPP). The intensity function of the latter, at any instant t, is given as the realization of a real nonnegative stochastic process $\{Y(t), 0 \le t \le T\}$. Denote by C_t the Borel field generated by $\{Y(\tau), 0 \le \tau \le t\}$. Then $\{N(t), 0 \le t \le T\}$ is an integer-valued process with independent increments, given a realization Y_T of $Y(\tau)$ over $[0,T]$, with a Poisson counting distribution; i.e.,

$$P\{N(t) = n | C_t\} = (n!)^{-1} \left(\int_o^t Y(u)\ du \right)^n \exp\left(- \int_o^t Y(u)\ du \right) . \quad (19)$$

Furthermore, we assume here $\{Y(t)\}$ to be a second-order positive piecewise-continuous process. Consequently, $\{N(t)\}$ is a regular point process (r.p.p.) [1], with (conditioned on Y_t) the intensity function

$$\lambda_{N(t)}(t, \mathscr{B}_t, Y_t) \equiv Y(t) \quad . \quad (20)$$

We consider a binary detection problem. A sample function of the counting process $\{N(t), 0 \le t \le T\}$ is observed, and one has to decide as to the existence of $\{Y^{(1)}(t)\}$ or $\{Y^{(2)}(t)\}$. The likelihood-ratio processor is expressed by (4) when intensity $\hat{\lambda}_n^{(i)}(t, t_n, \ldots, t_1)$ is replaced by

$$\hat{Y}_n^{(i)}(t, t_n, \ldots, t_1) = E\{Y^{(i)}(t) | N(t) = n, W_n = t_n, \ldots, W_1 = t_1\},\ i=1,2. \quad (21)$$

We note that in the case of optical communication systems, we have

$$Y(t) = \alpha |s(t)|^2 ,$$

where s(t) is the complex envelope of the received electric field, and α is related to the quantum efficiency of the photo-detector and the energy per photon at the carrier frequency. The stochastic behavior of Y(t) is caused by message modulation and fading of the optical field.

The best m-memory DSPP intensity is readily shown to be given by

$$\lambda_n(t,t_n,t_{n-1},\ldots,t_{n-m+1}) = -\frac{\partial}{\partial t} \ell n\ E\left\{ \prod_{i=n-m+1}^{n} Y(t_i) \left[\int_o^{t_{n-m+1}} Y(u)du \right]^{n-m} \exp\left[\int_o^t Y(u)du \right] \right\}, \quad (22)$$

for $n \ge m > 0$. The best 0-memory intensity is given by

$$\lambda_n(t) = \frac{E\left\{Y(t)\left[\int_o^t Y(u)du\right]^n \exp\left[-\int_o^t Y(u)du\right]\right\}}{E\left\{\left[\int_o^t Y(u)du\right]^n \exp\left[-\int_o^t Y(u)du\right]\right\}} . \tag{23}$$

The optimal m-memory detection scheme is then given by (4) with intensities (22)-(23) incorporated.

Using expressions (22)-(23) one obtains that a DSPP is 1-memory iff

$$E\left\{\exp\left[-\int_s^t Y(u)du\right]\Big|C_s\right\} = E\left\{\exp\left[-\int_s^t Y(u)du\right]\right\}, \quad \forall t > s > 0 \tag{24}$$

and is 0-memory iff $Y(t) = Y$ a random variable. The corresponding intensities in the above two cases are then:

$$\lambda_n(t,t_n) = -\frac{\partial}{\partial t}\, \ell n\, E\left\{\exp\left[-\int_{t_n}^t Y(u)du\right]\right\} ; \tag{25a}$$

$$\lambda_n(t) = \frac{E\{Y^{n+1}e^{-Yt}\}}{E\{Y^n e^{-Yt}\}} = \frac{\Phi_Y^{(n+1)}(t)}{\Phi_Y^{(n)}(t)} , \tag{25b}$$

where $\Phi_Y(u) = E\{e^{-Yu}\}$ is the generating function of Y, and $\Phi_Y^{(n)}(u) = \frac{d^n}{du^n}\Phi_Y(u)$.

Expression (24) (and its generalization to m > 1 [2]) indicates that Y(t) must have appropriate independence characteristics for the DSPP to have a reduced-memory character. In particular, the 0-memory and 1-memory cases are of practical interest. For the first $Y(t) \equiv Y$ and the resulting processor follows from (4) and (25b). Extending to the case where Y(t) is a piecewise constant R.V. one obtains [2] a generalized Reiffen-Sherman [9] test. To obtain, in general, a 1-memory processor one will have to approximate the incoming DSPP by a Markov $\{W_n\}$ sequence with a useful transition density such as the Gamma density [3], [2]. The resulting processors are then simple to implement while appropriately incorporate the recent past into the present processing operation. Simple processors also result when homogeneity assumptions concerning the Y(t) process can be made [3], [2].

REFERENCES

[1] I. Rubin, "Regular Point Processes and Their Detection," IEEE Trans. on Information Theory, Vol. 18, pp. 547-557, September 1972.

[2] I. Rubin, "Reduced Memory Likelihood Processing for Point Processes," submitted for publication.

[3] I. Rubin, "Detection of Point Processes and Applications to Photon and Radar Detection," Technical Report No. 32, Information Sciences and Systems Laboratory, Dept. of Electrical Engineering, Princeton University, September 1970.

[4] A.T. Bharucha-Reid, Elements of the Theory of Markov Processes and Their Applications, New York: McGraw-Hill, 1960, Chapter 2.

[5] E. Parzen, Stochastic Processes, San Francisco, California: Holden-Day, 1962.

[6] D.R. Cox, Renewal Theory, Methuen, 1962.

[7] D.L. Snyder, "Filtering and Detection for Doubly-Stochastic Poisson Processes," IEEE Trans. on Information Theory, pp. 91-102, January 1972.

[8] L. Mandel and E. Wolf, "Coherence Properties of Optical Fields," Rev. of Mod. Phys., Vol. 37, pp. 231-287, April 1965.

[9] B. Reiffen and H. Sherman, "An Optimum Demodulator for Poisson Processes: Photon Source Detectors," Proc. IEEE, Vol. 51, pp. 1316-1320, October 1963.

GEOMETRY OF QUANTUM SIGNAL DETECTION*

ROBERT O. HARGER AND MYONG C. KIM
Department of Electrical Engineering
University of Maryland
College Park, Maryland 20742

Quantum signal detection theory was formulated by Helstrom (1) and summarized in Helstrom, Liu and Gordon (2). In quantum signal detection problems, optimum solution is to find an observable (Hermitian operator), say X, such that measurement of X specify the structure of optimum receiver and decision rule. Under the probability of error criterion, we interpret the probability of correct decision as an inner product between two vectors so that we can treat the minimization of the probability of error geometrically. Following the notation in (2) for convenience, the probability of error P_e for M-ary detection problem is given by

$$P_e = 1 - \sum_n \sum_{j=1}^{M} \zeta_j P_{jn} \langle x_n | \rho_j | x_n \rangle \tag{1}$$

For minimum probability of error, P_{jn} was chosen either 1 or 0 depending on particular pair jn .

Thus let $$J_j \triangleq \sum_n P_{jn} \langle x_n | \rho_j | x_n \rangle = \sum_n I_{N_j}(n) \langle x_n | \rho_j | x_n \rangle \ , \tag{2}$$

where $I_{N_j}(n) = 1$, $n \in N_j$
$= 0$, otherwise.

Thus $$P_e = 1 - \sum_j \zeta_j J_j = 1 - (\vec{\zeta}, \vec{J}) \ , \tag{3}$$

where $\vec{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_M)$, $J = (J_1, J_2, \ldots, J_M)$.

Since $\vec{\zeta}$ is the fixed a-priori probability vector of hypotheses, minimum probability of error is achieved by finding a vector $\vec{J}$ which has maximum projection along the vector $\vec{\zeta}$. Hence optimum detection problem is reduced to the problem of finding the locus of $\vec{J}$. In general M-ary detection problems, determination of locus of $\vec{J}$ is quite involved. In the following, we shall consider a binary detection problem in 2-dimensional Hilbert space.

Let density operators ρ_1 and ρ_2 for hypothesis H_1 and H_2 respectively are given by

$$\rho_1 = \lambda_1^{(1)} |\phi_1\rangle\langle\phi_1| + \lambda_2^{(1)} |\phi_2\rangle\langle\phi_2| \ , \ \lambda_1^{(1)} + \lambda_2^{(1)} = 1$$
$$\rho_2 = \lambda_2^{(2)} |\psi_1\rangle\langle\psi_1| + \lambda_1^{(2)} |\psi_2\rangle\langle\psi_2| \ , \ \lambda_1^{(2)} + \lambda_2^{(2)} = 1 \tag{4}$$

---

*Research supported by NASA Grant NGR-21-002-237 and NSF Grant GK-14920.

Since orthonormal bases are unitarily related, we have the following equations:

$$
\begin{aligned}
|x_1\rangle &= a\,|\phi_1\rangle + \sqrt{1-a^2}\,|\phi_2\rangle \\
|x_2\rangle &= -\sqrt{1-a^2}\,|\phi_2\rangle + a\,|\phi_2\rangle \\
|\phi_1\rangle &= b\,|\psi_1\rangle + \sqrt{1-b^2}\,|\psi_2\rangle \\
|\phi_2\rangle &= -\sqrt{1-b^2}\,|\psi_1\rangle + b\,|\psi_2\rangle\,, \quad -1 \le a\,,\ b \le 1\,.
\end{aligned}
\tag{5}
$$

where $|x_1\rangle$ and $|x_2\rangle$ are eigenvectors of X.

The possible index sets are given as follows:

(i) $N_1 = \{1,2\}$, $N_2 = \emptyset$, (ii) $N_1 = \{1\}$, $N_2 = \{2\}$, (iii) $N_1 = \{2\}$, $N_2 = \{1\}$, (iv) $N_1 = \emptyset$, $N_2 = \{1,2\}$.

Using the relations in Eq. (5) and various index sets, we can determine the locus of $\vec{J}$. Except trivial cases, the locus of $\vec{J}$ turns out to be an ellipse. With a given $\vec{\zeta}$, a $\vec{J}$ is determined which has maximum projection along $\vec{\zeta}$. Eigenvectors of X, $\{|x_i\rangle\}$, are then determined, which in turn specify the optimum detection operator.

If we use the randomized decision strategy, i.e., $0 \le P_{jn} \le 1$, to include all possible values of probability of error, the locus of $\vec{J}$ becomes a convex region. Optimum binary detection problem with minimax criterion can also be solved using this geometrical method. Optimum solution in this case employ randomized decision strategy.

Further this geometrical method can be applied for the optimum solution of binary detection problem described in a product Hilbert space. For an example, let us consider a binary detection problem in product space with density operators ρ_1 and ρ_2 for hypothesis H_1 and H_2 respectively as follows:

$$
\begin{aligned}
\rho_1 &= \rho_{11} \otimes \rho_{12} \\
\rho_2 &= \rho_{21} \otimes \rho_{22}
\end{aligned}
$$

such that $\rho_{11}\rho_{21} = \rho_{21}\rho_{11}$, $\rho_{12}\rho_{22} \neq \rho_{22}\rho_{12}$.

We also assume that each Hilbert space is 2-dimensional. For optimum detection, we must find a pair of Hermitian operators $(X_1 \otimes I,\ I \otimes X_2)$ such that optimum decision depends on a pair of outcomes of measurement (X_{1i}, X_{2j}). Following the analogous procedures as shown in the above example, optimum observables are found as a pair of operators in which X_1 turns out to be commutative with ρ_{11}. This result is something we anticipated in view of the particular choice of the problem. In the earlier formulation of the binary detection problem (2), optimum detection operator was given by the solution of the eigenvalue equation. However the method of eigenvalue equation can not be applied when the problem is described in a product Hilbert space.

References

1. C. W. Helstrom, "Detection Theory and Quantum Mechanics," Information and Control, Vol. 10, No. 3, March 1968.
2. C. W. Helstrom, J. W. S. Liu and J. P. Gordon, "Quantum Mechanical Communication Theory," IEEE Proceedings, Vol. 58, No. 10, Oct. 1970.

APPROXIMATELY OPTIMUM DETECTION OF LINEAR PROCESSES

LESTER F. EASTWOOD, JR. and ROBERT LUGANNANI
Department of Electrical Engineering
Princeton University, Princeton, New Jersey 08540

ABSTRACT

This paper develops approximately optimum detectors for problems in which the signal or noise is a linear process. We represent the Bayes test function by a convergent series approximation and use a truncated version of the series to make the decision. The decision is optimum whenever the series approximation is exact. We apply the technique in several cases which model reverberation, multipath, and other detection problems.

I. INTRODUCTION

The problem of the optimum detection of signals in noise has been solved completely only under stringent assumptions. Most investigators assume that the noise and the signal, if it is stochastic, are Gaussian [1], [2]; this situation is one of the few for which an optimum detector can be implemented [3]. However, in some important practical cases, the Gaussian assumption is invalid [4], and using the Gaussian optimum detector risks unexpectedly poor performance.

Though nonparametric detectors eliminate this risk, we still seek detectors that are optimum for non-Gaussian problems. One motivation for this effort is that the optimal performance is a standard of comparison for that of simply implemented, suboptimum designs. Moreover, there may be problems for which only optimum performance is acceptable.

Many people have worked to solve the stochastic signal in noise problem. Most notably, Kailath [5] describes the optimum detector for a broad class of random signals in white Gaussian noise. Unfortunately, implementation of this detector requires the causal, mimimum mean-squared error estimator for the signal, and the structure of this device is usually unknown if the signal is nongaussian. Choosing a different approach, Schwartz [6] describes a Bayes optimum detection algorithm for the one-sample problem. This algorithm assumes that a convergent series expansion for the test statistic is known; it makes optimum decisions based on a truncated version of this series.

This paper develops approximately optimum detectors, based on Schwartz's algorithm, for detection problems in which the received random processes are linear processes, a class more general than the Gaussian. Using a method for approximate inversion of a characteristic function [7], we derive convergent series approximations for the probability density of a r.v. from a suitably restricted class of linear processes. Applying this expansion, we develop series approximations of the Bayes optimum test statistics for a variety of problems, including models of signal detection in narrowband reverberation noise, multipath communication, and other examples of practical value. In each case, moreover, we give a bound on the error caused by truncating the series approximation of the statistic. This bound permits the use of a truncated version of the series to make the decision, so the design is implementable. Whenever the series approximation of the test statistic is exact, our detector's decisions are Bayes optimum. Finally, we present computer results for each problem, including probabilities of error and of making a decision using a given number of terms of the series, and plots of approximate densities.

Two considerations motivate our use of the linear random process in this investigation. First, this class of processes includes many of those used to model physical phenomena. For example, Gaussian, Poisson, shot noise, and filtered compound Poisson processes are linear processes. So, too, are Kennedy's [1] and Middleton's [8] models of the random received waveforms in a scattering channel. Second, in addition to its value as a model in detection theory, the linear process has a property which enables us to attempt optimum detector design. Specifically, we know the finite-dimensional characteristic functions of the process [9], and therefore, we can sometimes approximate the required probability densities.

II. THE LINEAR PROCESS

In this section we present a general description of the linear process. This description is brief and is only for the purpose of discussing terminology; a more careful exposition appears in [9], which also lists further references. In addition, this section specifies the class of linear processes we use.

We represent the linear processes by stochastic integrals of the form

$$y(t) = \int_a^b f(t,s)\, dx(s) \tag{1}$$

for all t in a set T and for $-\infty \le a < b \le \infty$. The process x(s) is a zero-mean, second order, independent increment process which is continuous in probability. In addition, for each t in T, f(t,s) is real-valued, and square integrable in a special sense [9], so that (1) is well-defined as a l.i.m. of Stieltjes sums. Briefly stated, a linear process is an L_2 filtering of an independent increment process.

The linear processes retain two of the useful properties of the Gaussian and shot noise processes, of which they are a generalization. First, the class of linear processes is closed under linear transformations. Second, and important for our work, we know the finite dimensional characteristic functions of the process in (1); they are

$$\Phi_{\underline{t}}(\underline{u}) = \exp\left\{\int_a^b \int_{-\infty}^{\infty} \left[\exp(i\lambda w) - i\lambda w - 1\right]\left(1/\lambda^2\right) d_\lambda\, d_s G\left(\lambda, s\right)\right\} \tag{2}$$

where $\underline{t} = t_1, \ldots, t_n$; $\underline{u} = u_1, \ldots, u_n$; and $w = u_1 f(t_1,s) + \ldots + u_n f(t_n,s)$. The function $G(\lambda, s)$ is monotone, nondecreasing in both λ and s and is bounded in λ. Moreover, G is determined uniquely from the process x(t) by

$$-\frac{d^2}{du^2}\,\ell n E\Big[\exp\{iu[x(s)-x(s')]\}\Big]=\int_{-\infty}^{\infty}\exp(iu\lambda)\,d_\lambda\Big[G(\lambda,s)-G(\lambda,s')\Big]$$

A function related to G(λ,s) is meaningful for linear processes with no Gaussian component. In linear processes of this type, x(s) is a compound Poisson process; its sample function increments occur at times determined by a Poisson law, and the size, λ, of an increment is a r.v. with distribution F(λ,s) at time s. The "jump distribution", F(λ,s) is related to G(λ,s) by

$$G(\lambda,s) = \int_{-\infty}^{\lambda} z^2\, d_z\, F(z,s) \tag{4}$$

For example, when x(t) is a stationary, centered Poisson process with jumps of unit magnitude, F(λ,s) = U(λ-1), where U is the unit step.

We specialize the class of linear processes used in the remainder of the paper by one assumption on their nongaussian component. Let the r.v.λ, the "jump" amplitude, be the product of two independent r.v., r and cos θ, and assume that θ is uniformly distributed on [0,2π]. This restriction requires that the r.v.λ be symmetrically distributed about λ=0. Incidentally, most models of narrowband reverberation noise and of multipath signals satisfy this assumption.

This assumption on the jump distribution allows us to rewrite the ch.f. in (2). As a first step, we substitute $d_\lambda d_s F(\lambda,s)$ for $(1/\lambda^2)d_\lambda d_s G(\lambda,s)$, according to (4), in equation (2). We then recognize the inner integral in (2) as the expectation over λ, at time s, of the quantity, exp(iwλ)-iwλ-1. In addition, the fact that $\lambda = r\cdot\cos\theta$ permits us to take this expectation over r and θ. Finally, since r and θ are independent, their joint density $d_r d_\theta\, F(r,\theta;s)$ factors as $d_r F(r,s) d_\theta F(\theta,s)$: thus, it is $(1/2\pi)d_r F(r,s)$ on $0\leq\theta<2\pi$ and zero elsewhere. Applying this analysis, we obtain the ch.f. for the nongaussian component of our specialized linear process:

$$\Phi_{\underline{t}}(\underline{u})=\exp\left\{\int_{a}^{b}\int_{-\infty}^{\infty}\int_{0}^{2\pi}\frac{1}{2\pi}\Big[\exp(iwr\cos\theta)-iwr\cos\theta-1\Big]d\theta\, d_r d_s F(r,s)\right\}$$

or, taking the integral over θ,

$$\Phi_{\underline{t}}(\underline{u})=\exp\left\{\int_{a}^{b}\int_{-\infty}^{\infty}\Big[J_0(wr)-1\Big]d_r d_s F(r,s)\right\} \tag{5}$$

III. THE DETECTION ALGORITHM

This section describes the Bayes detection algorithm we use. This algorithm is due to Schwartz, and further discussion and some applications to stochastic signals in Gaussian noise may be found in his paper [6].

Consider a two-hypothesis, single observation detection problem. Under hypothesis H_i, $i=1,2$, the density of the observation, y, is $f_i(y)$. The a priori probabilities of H_1 and H_2 are p_1 and $p_2=1-p_1$, respectively. The test procedure which minimizes the total probability of error evaluates the likelihood ratio, $L(y)$, and compares it to one:

$$L(y) = p_2 f_2(y)/p_1 f_1(y) \underset{H_1}{\overset{H_2}{\gtrless}} 1 \tag{6}$$

An equivalent procedure evaluates the test statistic $T(y)$:

$$T(y) = p_2 f_2(y) - p_1 f_1(y) \underset{H_1}{\overset{H_2}{\gtrless}} 0 \tag{7}$$

Assume now that the densities have pointwise convergent series representations:

$$\left| f_i(y) - \sum_{m=0}^{q} \alpha_{im} \varphi_{im}(y) \right| < B_i(q,y) \;, \; i=1,2$$

where, as $q \to \infty$, the truncation error bound $B_i(q,y) \to 0$ for all y. Then (7) becomes

$$T(y) = \sum_{m=0}^{\infty} \left[p_2 \alpha_{2m} \varphi_{2m}(y) - p_1 \alpha_{1m} \varphi_{1m}(y) \right] \underset{H_1}{\overset{H_2}{\gtrless}} 0 \tag{8}$$

Define a truncated test statistic, $T_q(y)$, by truncating the series in (8) at the q^{th} term. The error bound, $B(q,y)$ for $T_q(y)$ is simply:

$$|T(y) - T_q(y)| \le p_1 B_1(q,y) + p_2 B_2(q,y) \equiv B(q,y) \tag{9}$$

An algorithm for Bayes optimal detection using $T_q(y)$ and $B(q,y)$ is

(a) $q=1$.

(b) calculate $T_q(y)$ and $B(q,y)$.

(c) if $|T_q(y)| > B(q,y)$, so that $\text{sgn}[T_q(y)] = \text{sgn}[T(y)]$, then announce H_2 if $T_q(y) > 0$; H_1 otherwise, and stop.

(d) if $|T_q(y)| < B(q,y)$, set $q=q+1$, and go to (b).

Since as $q \to \infty$, $B(q,y) \to 0$, the algorithm will always terminate.

The next section of this paper obtains convergent series representations of linear process densities and truncation error bounds for these series. The section following applies the algorithm above to these functions to design detectors.

IV. SERIES APPROXIMATIONS OF LINEAR PROCESS DENSITIES

In deriving the series approximations, we consider separately the linear processes with and without a Gaussian component, and we assume that the nongaussian component has a ch.f. of the form (5). Our derivations generalize a method of Middleton [7].

Our initial goal is to approximate the probability density of $y(t)$, the value at time t of a linear process with no Gaussian component. We may express this density in terms of the corresponding ch.f. by the Fourier inversion formula,

$$f(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_t(u) \exp(-iuy)\,du \tag{10}$$

To approximate the density, we seek an invertible approximation of the ch.f. $\Phi_t(u)$, which is (5) with $w=uf(t,s)$.

As a first step in this direction, we write the Bessel function in (5) in its series form [10]:

$$J_o(ruf(t,s)) = \sum_{k=o}^{\infty} (-1)^k (ruf(t,s)/2)^{2k}/(k!)^2 \tag{11}$$

By Fubini's theorem, we may do the integration over r and s in (5) term-by-term. We now define $\int_a^b \int_{-\infty}^{\infty} 1\, d_r d_s F(r,s) \equiv K$. Then, denoting the operation, $\int_a^b \int_{-\infty}^{\infty} [\cdot] d_r \bar{F}(r,s)/K$, by the brackets, $\langle\cdot\rangle$, and calling $z=rf(t,s)$, we have:

$$\langle J_o(u^2)\rangle = \langle 1\rangle - u^2\langle z^2\rangle/4 + u^4\langle z^4\rangle/64 - \ldots \tag{12}$$

Dividing the series (12) by the Taylor series for $\exp(-u^2\langle z^2\rangle/4)$ and multiplying the quotient by this quantity, we get

$$\begin{aligned} K\langle J_o(uz)-1\rangle = K\Big[&1+(u^4/64)(\langle z^4\rangle-2\langle z^2\rangle)-(u^6/2304)(\langle z^6\rangle-9\langle z^2\rangle\langle z^4\rangle \\ &+ 12\langle z^2\rangle^3+0(u^8)\Big] \times \Big[\exp(-u^2\langle z^2\rangle/4)\Big] - K \end{aligned} \tag{13}$$

This expression for the exponent of the ch.f. is exact when all correction terms are included. However, we can evaluate the transform, (10), if we ignore the correction terms in (13) of $O(u^4)$ and higher. This approximation under the integral is justifiable by the reasoning used in the steepest descent method of approximating an integral; the terms of $O(u^4)$ are small near the origin, and outside this region, the exponential becomes very small. With this approximation in (13), we obtain, by (5) and (10),

$$f(y) \cong \frac{1}{2\pi} \exp(-K) \int_{-\infty}^{\infty} \exp\Big[-iuy+K\cdot\exp(-u^2\langle z^2\rangle/4\Big]du \tag{14}$$

Evaluating (14) is a matter of two steps. First, we replace $\exp(K\cdot\exp(-u^2\langle z^2\rangle/4)$ by its Taylor series expansion. Then, interchanging the order of summation and integration, we transform this series term-by-term. The result is

$$f(y) \cong \exp(-K)\Big[\delta(y) + \sum_{m=1}^{\infty} \frac{K^m}{m!} \frac{\exp(-y^2/m\langle z^2\rangle)}{\sqrt{\pi m\langle z^2\rangle}}\Big] \tag{15}$$

We note that K, defined above as $\int_a^b\int_{-\infty}^{\infty} d_r d_s F(r,s)$ is the expected number of events (increments of the process $x(s)$) in the time interval (a,b) [11]. Moreover, using (15) to find the variance, σ_L^2, of the r.v. y, we obtain

$$\sigma_L^2 = \exp(-K) \sum_{m=1}^{\infty} \frac{mK^m}{m!} \cdot \frac{\langle Ƶ^2 \rangle}{2} = K \langle Ƶ^2 \rangle/2 \qquad (16)$$

Therefore, in the remainder of the paper, we shall substitute $2\sigma_L^2/K$ for $\langle Ƶ^2 \rangle$, in order to state our equations in the most physically meaningful way.

A plot of (15) for $\sigma_L^2=1$ and for various values of K appears in Fig. 1. The coefficient of the delta function for each K is tabulated on the figure. We can justify the appearance of this graph by our physical intuition; when K is small, x(s) only rarely has an increment in (a,b), so dx(s) and therefore y(t) is most likely zero. The increments, when they do occur, must be of large amplitude to make $\sigma_L^2=1$. As K increases, x(s) becomes the sum of many independent, small amplitude increments; therefore y(t) looks Gaussian.

For use in the detection procedure, we must derive a truncation error bound for the series in (15). The error in truncating the series at the q-th term is simply the sum of all terms with indices greater than q. Moreover, since the exponential in each of these terms is not greater than one, and since the quantity under the square root increases with m, a valid bound is

$$B(q,y) = \left[\exp(-K)/\sqrt{2\pi(q+1)\sigma_L^2/K}\right] \sum_{m=q+1}^{\infty} K^m/m!$$

Finally, by Taylor's remainder theorem [12], the sum in this equation is bounded above by $\exp(K) \cdot K^{q+1}/(q+1)!$. Therefore,

$$B(q,y) = K^{(q+1)}/\left[(q+1)! \cdot \sqrt{2\pi(q+1)\sigma_L^2/K}\right] \qquad (17)$$

By the same methods, we may derive a series expansion and an error bound when y(t) has a Gaussian component of variance σ_g^2. In this case, the ch.f. of the r.v., y, is the ch.f. in (5) multiplied by $\exp(-u^2\sigma_g^2/2)$, [11]. Multiplying the integrand in (14) by this quantity and proceding from (14) as before, we obtain

$$f(y) \cong \exp(-K) \sum_{m=o}^{\infty} \frac{K^m \exp(-y^2/D(m))}{m! \cdot \sqrt{\pi D(m)}} \qquad (18)$$

$$B(q,y) = K^{(q+1)}/\left[(q+1)! \cdot \sqrt{\pi D(q+1)}\right] \qquad (19)$$

where $D(m)=2(\sigma_g^2 + m\sigma_L^2/K)$ in these two equations, σ_L^2 is the variance of the nongaussian component of the r.v., y.

V. EXAMPLES OF DETECTOR DESIGN

This section designs approximately optimum detectors for three problems likely to appear in practice. For each problem, we employ the results of the last section to derive the detector functions $T_q(y)$ and $B(q,y)$. These functions, combined with the detection algorithm of section III, specify the detector. After analyzing each example, we present results of computer simulations of the detector.

Each of the three design problems has a physical interpretation which justifies its study. The first example, which is the detection of a known, additive signal in linear process noise with no Gaussian component, can model a radar target detection problem, since linear processes of this type may represent reverberation noise [1,8]. The second example is the detection of a known signal in linear process noise with a Gaussian component, and it models a radar or sonar problem in which receiver noise is important. The final example differs from the preceding two; it considers the detection of a stochastic signal in additive, independent noise. The signal is a linear process with no Gaussian component, while the noise is Gaussian. This structure may be interpreted as a model of multipath communications.

As stated, we consider first the detection of a known signal in nongaussian linear process noise. Equation (8) expresses the test statistic, T(y), in general terms. If we call H_1 the null (no signal) hypothesis, then (15) expresses $f_1(y)$ for this problem. Moreover, if S is the known signal level, then $f_2(y) = f_1(y-S)$. Therefore, by (8) and (15), $T_q(y)$ is

$$T_q(y) = \exp(-K) \sum_{m=1}^{q} \frac{K^m [p_2 \exp\{-(y-S)^2/D_I(m)\} - p_1 \exp\{-y^2/D_I(m)\}]}{m! \cdot \sqrt{\pi D_I(m)}} \quad (23)$$

where $D_I(m) = 2m\sigma_L^2/K$. We ignore the δ-functions in (15) since they do not affect the decision threshold. B(q,y), given in general by (9), is simply obtained; (17) expresses both $B_2(q,y+S)$ and $B_1(q,y)$, and since these functions are independent of y, B(q,y) also has form (17). Thus, equations (17) and (20), along with section III's algorithm, determine the detector. As a final step, we introduce as a parameter the signal-to-noise ratio (SNR) by defining it as $SNR=S^2/\sigma_L^2$. Computer simulations of this detector provide the results in Figs. 2 and 3. Fig. 2 shows detector performance as K and SNR change. In this graph, each data point results from 1000 trials at that K and SNR. Moreover, Fig. 3 shows the probability that the algorithm halts in q terms. Finally, note that Fig. 1 is a plot of $f_1(y)$ for various K.

The analysis of the second example, a known signal in linear process noise with a Gaussian component, resembles that above. Calling the variance of the nongaussian component σ_L^2, and of the Gaussian, σ_g^2, we define the signal level as $S=\sqrt{SNR(\sigma_L^2+\sigma_G^2)}$. Another parameter, $LGR=\sigma_L^2/\sigma_g^2$, determines the "mix" of the two noise components. Since (18) expresses $f_1(y)$ in this problem, $T_q(y)$, derived as in the first example, is

$$T_q(y)=\exp(-K) \sum_{m=0}^{q} \frac{K^m [p_2 \exp\{-(y-S)^2/D_{II}(m)\} - p_1 \exp\{-y^2/D_{II}(m)\}]}{m! \cdot \sqrt{\pi D_{II}(m)}} \quad (21)$$

where $D_{II}(m)=2(\sigma_g^2+m\sigma_L^2/K)$. Moreover, reasoning as before, (19) expresses B(q,y) for this problem. Fig. 4 shows the probability of error for LGR=0.1, $\sigma_L^2=1$, and various K and SNR. Simulations for LGR=1 and 10 produce graphs similar to Fig. 2.

The detection of a nongaussian, linear process signal in Gaussian noise is the final problem we consider. The signal component of the observation has variance σ_L^2, while the Gaussian part has zero mean and variance σ_g^2. Therefore, (18) expresses $f_2(y)$, the signal hypothesis density, and (19), $B_2(q,y)$. Moreover, $f_1(y)$ is the zero-mean Gaussian density with variance σ_g^2, so $B_1(q,y) = 0$, since we know f_1 exactly. Thus, by (8) and (9),

$$T_q(y)=p_2 \exp(-K)\sum_{m=0}^{q} \frac{K^m \exp(-y^2/D_{II}(m))}{m!\cdot\sqrt{\pi D_{II}(m)}} - \frac{p_1 \exp(-y^2/2\sigma_g^2)}{\sqrt{2\pi\sigma_g^2}} \qquad (22)$$

$$B(q,y)=p_2 K^{(q+1)}/[(q+1)!\cdot\sqrt{\pi D_{II}(q+1)}] \qquad (23)$$

Finally, we define $\mathrm{SNR}=\sigma_L^2/\sigma_g^2$. Probability of error results for various K and SNR appear in Fig. 5.

VI. DISCUSSION

We are extending this work in several directions. Most important, we are designing a multiple-observation analog of our detection scheme to obtain better performance. We have preliminary results on both multivariate density expansions and detection algorithms. In addition, we are working to evaluate the quality and range of validity of our density approximations; section IV's physical argument shows that they are reasonable, but we prefer a mathematical justification. Finally, we are designing methods to measure the parameters which appear in our equations; we must know, for example, the values of K and LGR to build the third detector in section V.

ACKNOWLEDGEMENT

The research reported in this paper was supported by the National Science Foundation under Grant GK-24187 and by the U.S. Army Research Office under Contract DAHC04-69-C-0012.

REFERENCES

1. R.S. Kennedy, Fading Dispersive Communication Channels, Wiley, New York, 1969.

2. H.L. Van Trees, Detection, Estimation, and Modulation Theory, Part III-Radar-Sonar Signal Processing and Gaussian Signals in Noise, Wiley, New York, 1971.

3. C.W. Helstrom, Statistical Theory of Signal Detection, Pergammon, New York, 1968.

4. G.V. Trunk, and S.F. George, "Detection of Targets in Non-Gaussian Sea Clutter", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-6, No. 5, September 1970, pp. 620-628.

5. T. Kailath, "A General Likelihood-Ratio Formula for Random Signals in Gaussian Noise", IEEE Transactions on Information Theory, Vol. IT-15, No. 3, May 1969, pp. 350-361.

6. S.C. Schwartz, "A Series Technique for the Optimum Detection of Stochastic Signals in Noise", IEEE Transactions on Information Theory, Vol. IT-15, No. 3, May 1969, pp. 362-369.

7. D. Middleton, "Probability Models of Received Scattered and Ambient Fields", Proceedings of the Sixth Princton Conference on Information Sciences and Systems, March 1972, pp. 66-71.

8. D. Middleton, "A Statistical Theory of Reverberation and Similar First-Order Scattered Fields", Part I, IEEE Transactions on Information Theory, Vol. IT-13, No. 3, July 1967, pp. 372-392.

9. R. Lugannani, and J.B. Thomas, "On a Class of Stochastic Processes which are Closed under Linear Transformations", Information and Control, Vol. 10, No. 1, January 1967, pp. 1-21.

10. F.B. Hildebrand, Advanced Calculus for Applications, Prentice Hall, Englewood Cliffs, N.J., 1962, p. 143.

11. J.L. Doob, Stochastic Processes, Wiley, New York, 1953, p. 422.

12. G.B. Thomas, Calculus and Analytic Geometry, Addison-Wesley, Reading, Mass., p. 796.

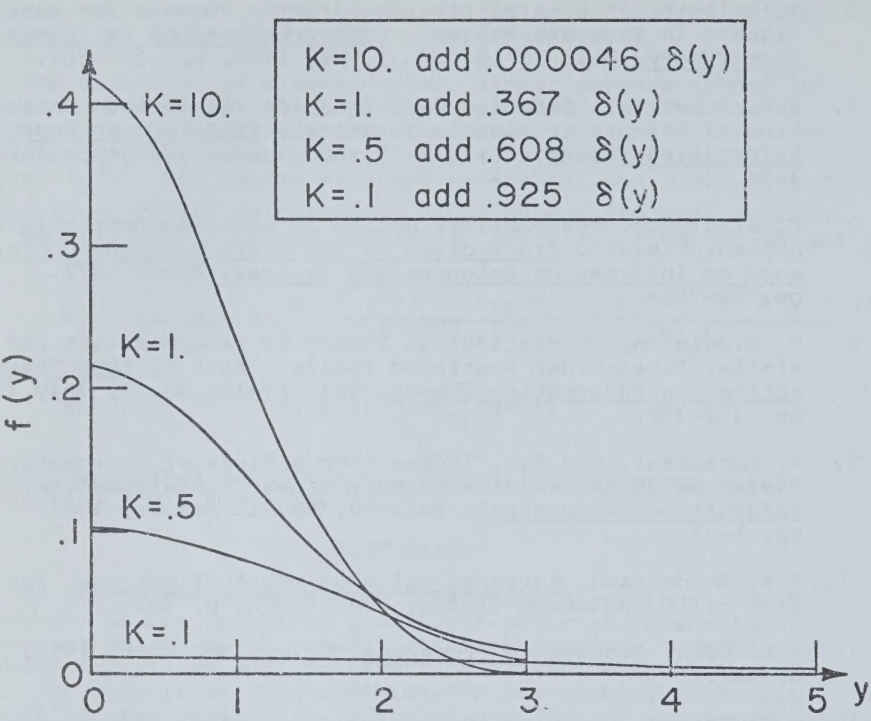


Fig. 1. Probability Density Approximation for Linear Process with No Gaussian Component. $\sigma_L^2 = 1$

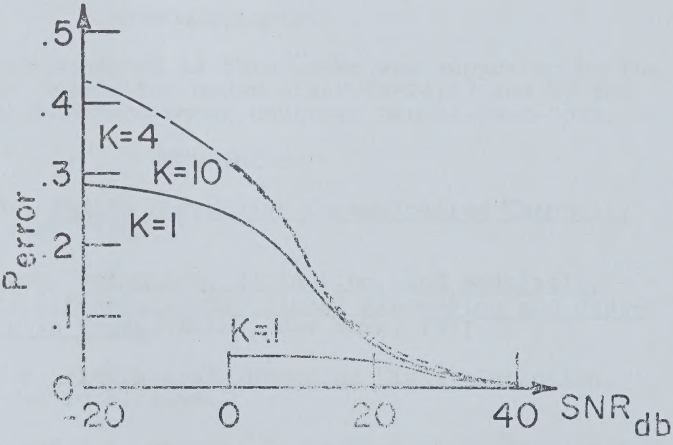


Fig. 2. Performance of Detector I. $\sigma_L^2 = 1$

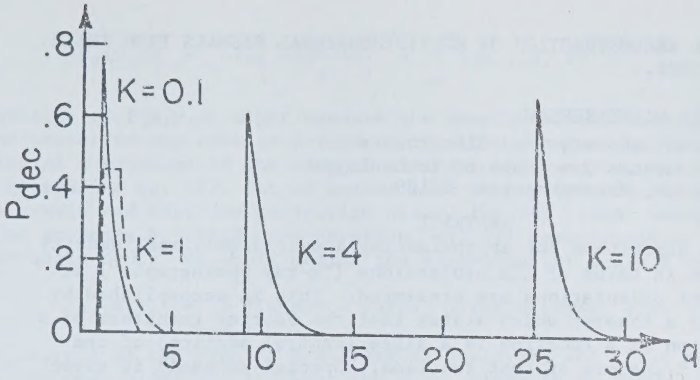


Fig. 3. Probability of Decision in q Terms. Detector I, $\sigma_L^2 = 1$

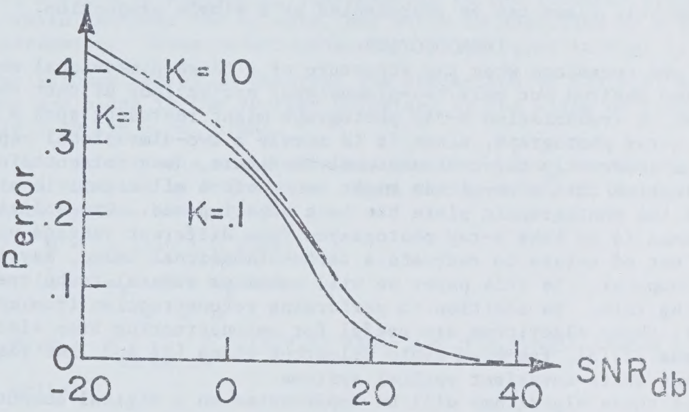


Fig. 4. Performance of Detector II. $\sigma_L^2=1$, LGR = 0.1

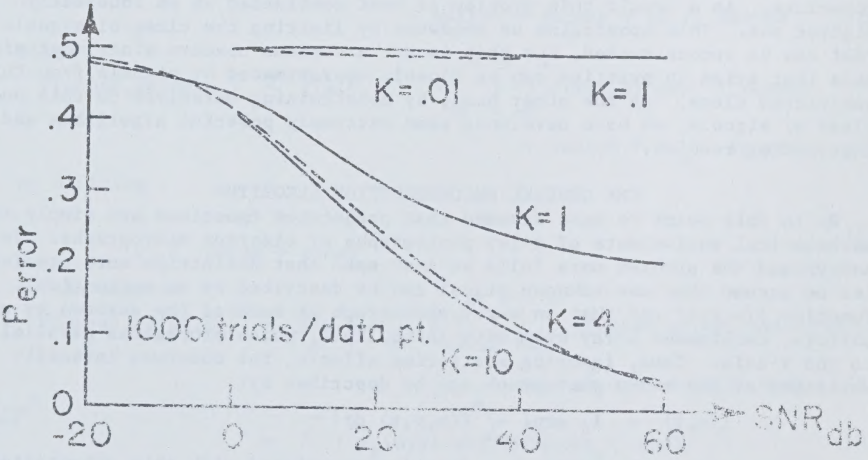


Fig. 5. Performance of Detector III. $\sigma_L^2 = 1$

THE DIGITAL RECONSTRUCTION OF MULTIDIMENSIONAL SIGNALS FROM THEIR PROJECTIONS.

RUSSELL M. MERSEREAU
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

Several algorithms for approximating a multidimensional density function in terms of its projections ("x-ray photographs") at different orientations are presented. This is accomplished by means of a theorem which states that the Fourier transform of a projection of a function is a slice (central section) of the Fourier transform of that function. Special emphasis is given to bandlimited functions as these are readily digitized. Some theorems are presented which state that bandlimited functions of a certain class can be represented by a single projection.

INTRODUCTION

There are occasions when the structure of a three-dimensional object is unknown and desired but only two-dimensional projections of that object are available. A transmission x-ray photograph might represent such a projection. A single x-ray photograph, since it is merely a two-dimensional representation of an inherently three-dimensional structure, does not contain all of the information that a physician might want, since all detail in a direction normal to the photographic plate has been superimposed. One solution to this dilemma is to take x-ray photographs from different vantage points and use this set of x-rays to recreate a three-dimensional image, say in a digital computer. In this paper we will consider several techniques for accomplishing this. In addition to performing reconstruction from x-rays [4][5][7][8], these algorithms are useful for reconstructing from electron micrographs [2][3], fan-beam radio telescope scans [1] and line responses from linear shift-invariant optical systems.

Because these algorithms will be implemented on a digital computer, we are constrained to work with sampled data. Our input projections must be sampled and our output reconstruction will consist of samples of the unknown structure. As a result this problem is best considered as an inherently digital one. This constrains us somewhat by limiting the class of signals that can be reconstructed, but this is not a serious concern since most signals that arise in practice can be closely approximated by signals from this restricted class. On the other hand, by constraining ourselves to this one class of signals, we have developed some extremely powerful algorithms and interesting results.

THE GENERAL RECONSTRUCTION ALGORITHM

Up to this point we have assumed that projection functions are simply the mathematical equivalents of x-ray photographs or electron micrographs. To understand the problem more fully we must make that definition more precise. Let us assume that one unknown signal can be described by an extinction function f(x,y,z) and that an x-ray photograph is made of the unknown by a uniform, collimated x-ray beam with intensity I_0 which propagates parallel to the y-axis. Then, ignoring scattering effects, the observed intensity variation of the x-ray photograph can be described by:

$$I(x,z) = I_0 \exp\{ -\int_{-\infty}^{\infty} f(x,y,z)\, dy\} \qquad (1)$$

We can then define the projection function associated with this orientation by

$$p_0(x,z) = -\ln\left[\frac{I(x,z)}{I_0}\right] = \int_{-\infty}^{\infty} f(x,y,z)\,dy \tag{2}$$

The function of $f(x,y,z)$ might measure the density of the unknown as it does to some extent in the case of x-rays or it might measure the extent of staining of a specimen in the case of electron micrographs, etc.

To generalize eq. (2), let us assume that each projection is taken normal to the z-axis and that the projection plane, the u-z plane, meets the x-z plane at an angle θ. With this notation, eq. (2) corresponds to the projection at angle $\theta=0^\circ$. We shall define the projection at angle θ by

$$p_\theta(u,z) = \int_{-\infty}^{\infty} f(u\cos\theta - v\sin\theta,\ u\sin\theta + v\cos\theta,\ z)\,dv \tag{3}$$

If $F(\omega_x,\omega_y,\omega_z)$ is the Fourier transform of $f(x,y,z)$, and if we define a plane, the ω-ω_z plane to intersect the ω_x-ω_z plane at an angle ϕ, then we can further define the slice of $F(\omega_x,\omega_y,\omega_z)$ at angle ϕ to be $F(\omega\cos\phi,\omega\sin\phi,\omega_z)$, i.e., a slice of $F(\omega_x,\omega_y,\omega_z)$ corresponds to the function evaluated on a plane, which includes the ω_z axis, and which is specified by a single angular parameter. These relationships are illustrated in Fig. 1.

Theorem: (projection/slice theorem) The projection of $f(x,y,z)$ at angle θ to the x-z plane is the Fourier transform of the slice of $F(\omega_x,\omega_y,\omega_z)$ at angle θ to the ω_x-ω_z plane.

Proof: Let us define a coordinate system, the $\omega,\hat{\omega},\omega_z$ coordinate system, which is a rotation by θ (radians) about the ω_z-axis, of the $\omega_x,\omega_y,\omega_z$ coordinate system.

$$\begin{aligned} \omega &= \omega_x\cos\theta + \omega_y\sin\theta \\ \hat{\omega} &= \omega_x\sin\theta + \omega_y\cos\theta \end{aligned} \tag{4}$$

Then since

$$F(\omega_x,\omega_y,\omega_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)\exp(-j[x\omega_x+y\omega_y+z\omega_z])\,dxdydz \tag{5}$$

we can write

$$\hat{F}(\omega,\hat{\omega},\omega_z) = F(\omega_x,\omega_y,\omega_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)\exp(-j[x\omega\cos\theta + y\omega\sin\theta - x\hat{\omega}\sin\theta + y\hat{\omega}\cos\theta + z\omega_z)]\,dxdydz\ . \tag{6}$$

Since

$$F(\omega\cos\theta,\omega\sin\theta,\omega_z) = \hat{F}(\omega,0,\omega_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)\exp(-j[x\omega\cos\theta + y\omega\sin\theta + z\omega_z])\,dxdydz \tag{7}$$

by defining

$$\begin{aligned} u &= x\cos\theta + y\sin\theta \\ v &= -x\sin\theta + y\cos\theta \end{aligned} \tag{8}$$

we can observe

$$F(\omega\cos\theta,\omega\sin\theta,\omega_z) = \hat{F}(\omega,0,\omega_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} f(u\cos\theta-v\sin\theta,\ u\sin\theta+v\cos\theta,\ z)\,dv\right\}\exp\{-j(u\omega+z\omega_z)\}\,dudz \tag{9}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_\theta(u,z)\exp\{-j(u\omega+z\omega_z)\}\,dudz \tag{10}$$

Q.E.D

This theorem forms the basis for our reconstruction algorithms. First we obtain as an input a set of projections (say from processed x-rays). It is most convenient if these are all taken normal to a single axis which we have defined as the z-axis in our coordinate system, although this is not necessary for the projection/slice theorem can be generalized. Then we transform these projections, by low pass filtering them, sampling them and performing a discrete Fourier transform (DFT) calculation. These samples of the Fourier transforms of the set of projections provide us with samples of $F(\omega_x,\omega_y,\omega_z)$ by virtue of the projection/slice theorem. From these we can estimate the entire Fourier space and then by inverse Fourier transforming in three dimensions, we have an estimate of $f(x,y,z)$, or, in our terminology, a reconstruction. How many projections are needed, how they should be sampled and how the Fourier space should be estimated from samples of the slices depend upon the individual algorithms.

SPECIFIC RECONSTRUCTION ALGORITHMS

Let us assume that the functions to be reconstructed are bandlimited. This assumption will guarantee that all of the projections are bandlimited and it will thus allow us to compute Fourier transforms from samples of the projections with no loss of information, a necessity for performing a reconstruction digitally. Also for our arguments let us assume that we are trying to reconstruct a two-dimensional function (picture) from one-dimensional projections. This is equivalent to reconstructing for only a single value of z.

One approach to use to obtain samples of the Fourier transform of an unknown picture is to sample each projection at the same sampling rate, that rate being greater than the maximum Nyquist rate of each of the projections. If then each sequence of samples is Fourier transformed using a DFT algorithm, the Fourier transform of the unknown picture $f(x,y)$ will be known on a polar raster of points. If the projections were evenly spaced from 0 to π, the raster of points is that shown in Fig. 2(a). From these samples of $F(\omega_x,\omega_y)$ there are a number of techniques that can be used to estimate $f(x,y)$. One of the more successful that we have used is to use linear interpolation to estimate the values of $F(\omega_x,\omega_y)$ on a Cartesian (square) raster from the polar samples. From the Cartesian raster we can perform an inverse two-dimensional DFT to obtain our estimate. Another approach that has yielded good reconstructions from a polar raster is to write the inverse transform integral in polar coordinates,

$$F(x,y) = \frac{1}{4\pi^2} \int_0^\infty \int_{-\pi}^{\pi} \hat{F}(\omega,\theta) \exp\{j(x\omega\cos\theta + y\omega\sin\theta)\}\omega \, d\omega d\theta \tag{11}$$

and then approximate (11) by a sum, where each summand depends upon one of the polar samples. The latter technique performs well resolving detail but does poorly on areas of low information content, such as backgrounds or areas of nearly constant grey level. The interpolation technique, on the other hand, does just the opposite. It resolves the flat areas but does not extract detail well.

As an alternative to sampling each projection at the same sampling rate we can vary the sampling rate with the projection angle. In particular, assume that $f(x,y)$ is bandlimited within a square in the frequency domain such that $F(\omega_x,\omega_y) \equiv 0$ for $|\omega_x| \geq W$ or $|\omega_y| \geq W$. Then the bandwidth of the projection at angle θ is defined by

$$W_\theta = \frac{W}{\max\{|\cos\theta|, |\sin\theta|\}} \tag{12}$$

Now suppose we sample each projection at a sampling rate which is proportional to its bandwidth. If these sequences are then DFT'd, $F(\omega_x,\omega_y)$ will be known on a concentric square raster such as that in Fig. 2(b). At first glance such a set of points is not to be favored over the polar set, but such is not the case. Using both the interpolation algorithm and the integral approximation algorithm, we get better reconstructions from the concentric squares raster than from the polar one. One reason for this can be seen from the fact that the Cartesian raster of samples which we must have in order to perform an inverse DFT lie along the same horizontal and vertical lines as the sides of the concentric squares. Thus instead of the necessity of performing a two-dimensional linear interpolation, we only need to perform a one-dimensional one. This might be expected to produce less error.

The second advantage to a concentric squares raster becomes apparent when we consider a special class of signals. We have assumed that f(x,y) is bandlimited. Let us now assume that it can be represented in the form

$$f(x,y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f\left(\frac{m\pi}{W},\frac{n\pi}{W}\right) \frac{\sin W(x-\frac{m\pi}{W}) \sin W(y-\frac{n\pi}{W})}{W^2 (x-\frac{m\pi}{W})(y-\frac{n\pi}{W})} \tag{13}$$

Thus in addition to requiring that f(x,y) be bandlimited, we have required that when sampled at its Nyquist rate it have only a finite number of non-zero samples. The number N which represents the width of the square of non-zero samples we will refer to as the order of f(x,y). Bandlimited functions of finite order are completely specified by their DFT's and their Fourier transforms are two-dimensional polynomials of degree N-1 in each variable. Because of the fact that a one-dimensional polynomial of degree d is completely specified by d+1 independent samples, it can be proven that a bandlimited function of order N can be reconstructed exactly from N+1 projections (although not necessarily for all sets of N+1 projections). Thus for this class of functions, concentric square projections as these projections shall be called, assume theoretical importance.

In Fig. 3 is shown a reconstruction of a cross section of a leg bone from 36 concentric squares projections. The projections were sections of x-rays which were taken at 5° intervals which are scanned logarithmically. Each section of each x-ray consisted of 256 samples and the reconstruction is plotted on a 256x256 Cartesian raster.

RECONSTRUCTING FROM ONE PROJECTION

In the previous section we noticed that bandlimited functions of finite order could be reconstructed exactly from N+1 (concentric squares) projections, where N was the order of the function. Is this the minimum number of projections needed to reconstruct these functions? The answer to this question is no. In fact, functions of this form can be reconstructed from a single projection.

Theorem: (one projection theorem) A bandlimited function f(x,y) of order N in each variable can be reconstructed from the concentric squares projection at $\theta = \tan^{-1}(1/N)$.

Proof: Consider the slice at $\theta = \tan^{-1}(1/N)$.

$$F(\omega\cos\theta,\omega\sin\theta) = F\left(\frac{\omega N}{\sqrt{N^2+1}}, \frac{\omega}{\sqrt{N^2+1}}\right) \tag{14}$$

where

$$F(\omega_x,\omega_y) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(\frac{m\pi}{W}, \frac{n\pi}{W}) \exp(-j[\frac{\pi}{W}(m\omega_x+n\omega_y)])\, b_{WW}(\omega_x,\omega_y) \qquad (15)$$

where

$$b_{WW}(\omega_x,\omega_y) = \begin{cases} 1, & |\omega_x| \leq W, \ |\omega_y| \leq W \\ 0, & \text{otherwise} \end{cases}$$

Evaluating (14),

$$F(\frac{\omega N}{\sqrt{N^2+1}}, \frac{\omega}{\sqrt{N^2+1}}) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(\frac{m\pi}{W}, \frac{n\pi}{W}) \exp\{-j(\frac{\pi\omega}{W\sqrt{N^2+1}}[Nm+n])\} \qquad (16)$$

$$|\omega| \leq \frac{W}{N}\sqrt{N^2+1}$$

If we define $g(Nm+n) = f(\frac{m\pi}{W}, \frac{n\pi}{W})$, then (16) becomes

$$F(\frac{\omega N}{\sqrt{N^2+1}}, \frac{\omega}{\sqrt{N^2+1}}) = \begin{cases} \sum_{p=0}^{N-1} g(p) \exp(-j\frac{\pi\omega p}{W\sqrt{N^2+1}}), & |\omega| \leq \frac{W}{N}\sqrt{N^2+1} \\ 0 & \text{otherwise} \end{cases} \qquad (17)$$

Thus over the region of interest, the slice at $\theta = \tan^{-1}(1/N)$ (which can be obtained from one projection) is a one-dimensional polynomial of degree N^2-1 in the variable $\exp\{-j(\pi\omega/W\sqrt{N^2+1})\}$, and the coefficients of that polynomial are simply the picture samples arranged column by column. Thus knowledge of N^2 sample values specifies the picture samples and by (13) this specifies the unknown picture function.

Q.E.D.

This theorem also has implications in two-dimensional filter design for it implies that the frequency response of a two-dimensional non-recursive digital filter is specified by its values along a single radial line, although how this fact might be utilized in filter design is still not clearly understood.

Despite the beauty of the one-projection theorem, it is not particularly useful as a reconstruction technique on a finite precision machine, because for values of N larger than 4 or 5, solving eq. (17) for $\{f(m\pi/W, n\pi/W)\}$ requires the inversion of a large nearly singular matrix and any errors made in obtaining the slice samples, which are bound to occur, are amplified enormously. In this respect, applying the one projection theorem is much like trying to apply analytic continuation to an unknown function all of whose derivations are known at a single point.

From a theoretical viewpoint, it is interesting to ask if there are any other of these critical slices. Clearly not all slices will work; for example, knowledge of sample values along just the ω_x or ω_y axis is generally not sufficient. In fact, there are an infinite number of these critical slices. A necessary and sufficient condition for a slice with a rational slope to be a critical slice is given in the following theorem which will be stated without proof.

Theorem: If a slice has an angle $\tan^{-1}(A/B)$ where A and B are integers with no common factor, then a necessary and sufficient condition for this slice to be sufficient for reconstruction of a bandlimited function of order N is that the equation

$$Bm + An = Bm' + An', \quad 0 \leq m,n,m',n' \leq N-1$$

possess only the trivial solution $\begin{array}{l} m = m' \\ n = n'. \end{array}$

In particular, if N is a power of two and A is an odd integer, the slice with slope $\tan^{-1}(A/N)$ is critical.

Using techniques completely analogous to that by which we derived the one projection theorem, we can derive two projection theorems, four projection theorems, etc. As a rule of thumb, the greater the number of projections which we care to use, the easier it is to obtain a solution with real data. It generally takes about N/2 projections for the sensitivity to be reduced enough to use currently available machines.

SUMMARY

A number of algorithms have been presented for reconstructing multidimensional signals from their projections, such as x-rays. In addition to their uses in performing reconstructions, projection functions are potentially useful for characterizing multidimensional signals for purposes of pattern recognition or bandwidth compression for signal transmission. Uses for projections in these areas has not been explored.

ACKNOWLEDGEMENT

This research was supported in part by the Advanced Research Projects Agency of the Department of Defense, monitored by ONR under Contract No. N0014-67-A-0204-0064, and by National Science Foundation Grant No. GK-31353.

REFERENCES

1. R.N. Bracewell, "Strip Integration in Radio Astronomy", *Austr J. Phys.*, 9:198, 1956.
2. R.A. Crowther, D.J. DeRosier and A. Klug, "The Reconstruction of a Three-Dimensional Structure from Projections and its Application to Electron Microscopy", *Proc. Roy. Soc. (Lond.) A*, 317:319, 1970.
3. D.J. DeRosier and A. Klug, "Reconstruction of Three-Dimensional Structures from Electron Micrographs", *Nature*, 217:130, 1968.
4. J.B. Garrison, D.G. Grant, W.H. Guier and R.J. Johns, "Three-Dimensional Roentgenography", *Am. J. Roentgenology, Radium Therapy and Nuclear Medicine*, 60:903, 1969.
5. R. Gordon, R. Bender and G.T. Herman, "Algebraic Reconstruction Techniques (ART) for Three-Dimensional Electron Microscopy and X-Ray Photography", *J. Theo. Biol.*, 29:471, 1970.
6. R.M. Mersereau, "Recent Advances in the Theory of Reconstructing Multi-Dimensional Signals from Projections". *MIT Res. Lab. Electron. Quart. Progr. Rept*. No. 105:169, 1972.
7. G.N. Ramachandran and A.V. Lakshminarayanan, "Three-Dimensional Reconstruction from Radiographs and Electron Micrographs III: Description and Application of the Convolution Method", *Indian J. Pure Appl. Phys*. 9:997, 1971.
8. O.J. Tretiak, M. Eden and W. Simon, "Internal Structure from X-Ray Images", 8th *ICMBE*, 1969.
9. B.K. Vainshtein, "Finding the Structure of Objects from Projections", *Sov. Phys. Crys.*, 15:781, 1971.

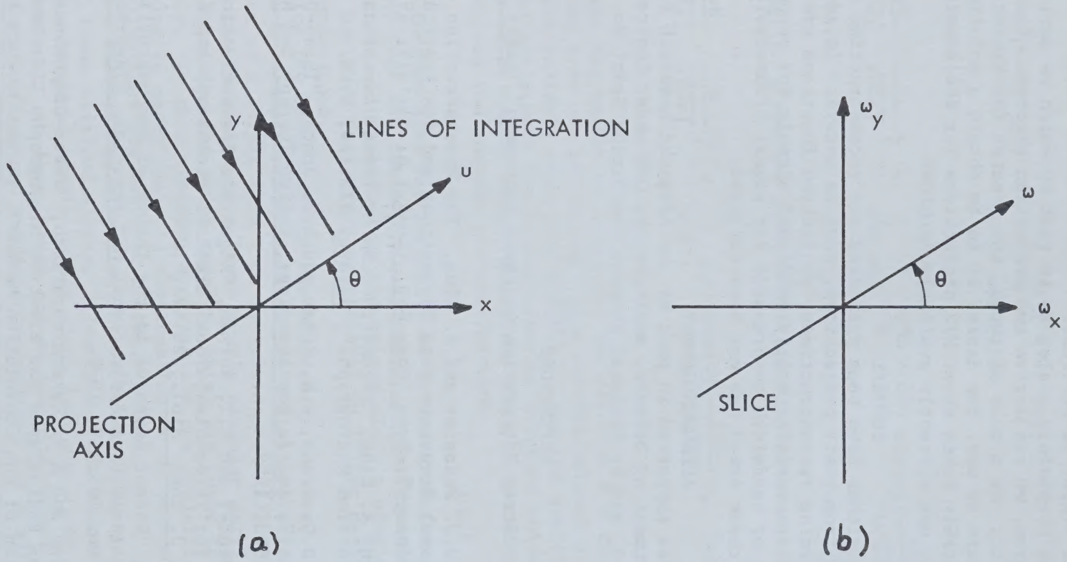

y
LINES OF INTEGRATION
u
θ
x
PROJECTION AXIS
(a)
ω_y
ω
θ
ω_x
SLICE
(b)

Figure 1 -- Definitions of (a) the projection of $f(x,y,z)$ at angle θ. The z-axis is normal to the page. (b) The slice of $F(\omega_x,\omega_y,\omega_z)$ at angle θ. The ω_z-axis is normal to the page.

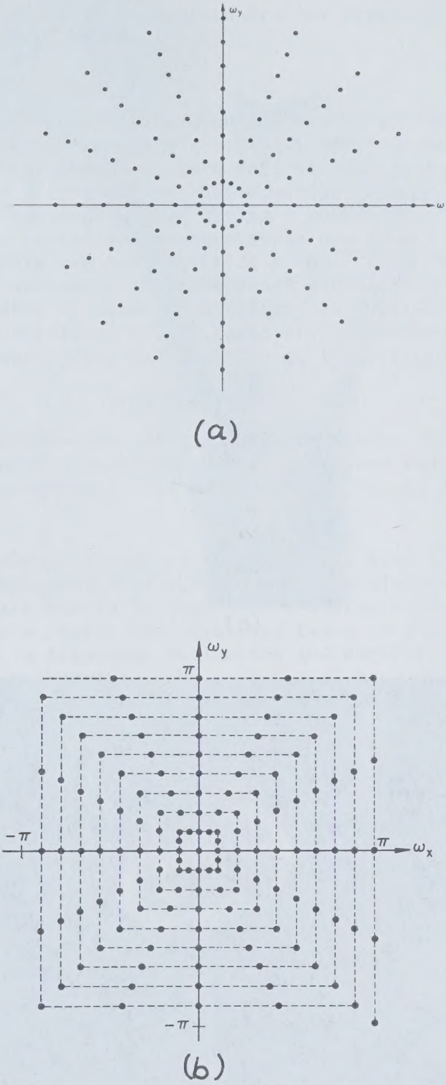


Figure 2 -- The samples of $F(\omega_x,\omega_y)$ that can be obtained by (a) sampling 8 projections all at the same rate, (b) sampling each of 8 projections at a rate proportional to its bandwidth.

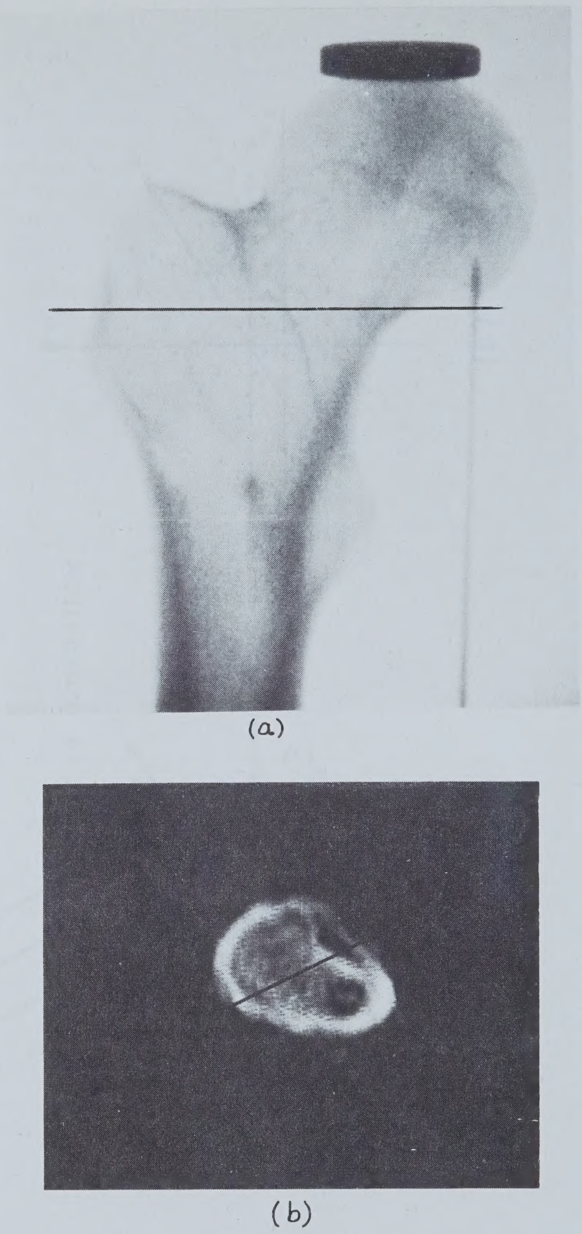


Figure 3 -- A reconstruction of a section of a human femur from 36 concentric squares projections at equally spaced angles. (a) One of the x-rays from which the reconstruction was made. (b) The reconstructed cross-section. Note: The two photographs are not to scale. The photgraph in (a) is a positive, that in (b) is a negative.

THE PRELIMINARY DESIGN OF A MULTICHANNEL COGNITIVE SIGNAL PROCESSOR AND SIMULATOR

J. M. BANDA
University of Minnesota, Minnesota Systems Research, Inc.
Minneapolis, Minnesota

ABSTRACT

In this paper a proposal is presented for the preliminary design of a cybernetic system which receives spatial or spatiotemporal signals from a given source and disperses them into various paths of its network where each signal is preprocessed. Subsequently, the preprocessed signals are made to converge, thus forming a composite pattern which is then processed or tested for membership to one of a number of problem areas - if this was not initially known. Once the general classification has been made, the integrated signal is transformed and spatially modulated by means of a filter and then either autocorrelated or crosscorrelated with signals stored in memory until its closest match is found. This match yields an identification or diagnosis.

I. INTRODUCTION

This paper describes some of the research I have done on the first stages of the design of a cognitive signal processor and simulator which I propose to construct. The research is still in its initial stages.

Because of my physiological slant, I have in mind to adapt the system to medical differential diagnosis and physiological research problems. My ultimate aim is to construct a system which will perform differential diagnosis, carry out cognitive tasks in physiological investigations, aid in treatment evaluation and perform other useful tasks in a medical environment. The system will be a physico-mathematical cybernetic automaton aimed at understanding and simulating the pattern recognition process of natural automata. It may also be viewed as an attempt at functional simulation of the cognitive information processing processes of the Central Nervous System with the intention of resolving the problem of cognition in natural systems.

In as far as it will be built around a mathematical conceptual model, essentially that proposed in my earlier published paper: A Mathematical Model for Computer Simulation of the Process of Differential Diagnosis in Medicine, it is evident that the value of the system in interpreting neuro-physiological phenomena will be limited. This limitation may be summed up by sounding a warning from one of Von Neumann's celebrated papers: "...it is dangerous to identify the real physical (or biological) world with the models which are constructed to explain it. The problem of understanding the animal nervous action is far deeper than the problem of understanding a computing machine. Even plausible explanations of nervous reaction should be taken with a very large grain of salt." It cannot be underemphasized, therefore, that the aim of the proposed system is a better understanding of C.N.S. function by simulation rather than simulation of the actual mechanisms of brain function.

By incorporating into it what I consider to be the salient features of pattern recognition, information processing and decision-making in biological systems, and by using the latest known techniques from electronics and optical data processing, I hope to come up with a signal processor and simulator capable of functioning in conjunction with, or independently of a computer. The system will have hybrid characteristics and in fact it will take a hybrid computer to simulate it. Eventually the system will be made compatible with standard data-collecting instruments from hospitals and medical research labs on an on-line or off-line basis and it may also be interfaced with optical equipment such as radiographic or fluoroscopic systems through an opto-electric link.

In this paper I am interested in discussing the organization of the system, its physical structure and its mathematical aspects.

In designing a cybernetic instrument, one important question keeps cropping up: Is it reasonable to simulate human thought by means of a machine? This is surely a difficult question to answer, for the nature of human thought is such that it self-destructs the moment it is reflected upon. Therefore, introspection is not valid as a research method for human cognitive thought. Moreover, the fact that the solitude of the mind-brain and individual thought remains impenetrable to outsiders does not help matters very much! One argument goes that computers or man-made systems cannot think because their processes are both accessible to outsiders and non-self-destructive, in distinct contrast to human thought. There may be some substance to this argument. But all-the-same I have chosen to steer clear of this issue for the time being and argue instead that the accessibility and replicability of computer process has finally brought with it a methodology for studying human thought, and in particular cognitive problem solving.

Therefore my motivation for proposing to construct the system described here is that studying the dynamic behavior of such a system lends insight into the dynamic behavior of the human mind that produces the object system behavior.

Lastly, I must mention that some of the methods proposed for processing the signals are similar to those used in optical data processing and pattern recognition. This is not a mere coincidence. It is a deliberate attempt on my part to exploit the strong analogies which appear to exist between optical data processors and nerve networks of biological systems: Both systems -

1) Perform spatial dispersion or transformation of input data into "interfere" or defocused patterns;
2) Perform parallel processing of multichannel data;
3) Are capable of modulating signals;
4) Have diverging and converging "networks";
5) and both recognize patterns by testing or comparing input patterns against patterns in memory.

II. A GENERAL VIEW OF THE SYSTEM

The model on which the system's organization is based relates symptoms x_j ($j = 1, 2, \ldots\ldots m$) and diseases z_i ($i = 1, 2, \ldots\ldots n$) as follows:

$$Z_{ij} = P_{ij} \sum_{j=1}^{m} w_{ij} x_j \quad , \; 0 \leq P_{kj} \leq 1 \; , \; 0 \leq W_{kj} \leq 1 . \qquad (1)$$

i.e., it defines diseases as convex combinations of certain symptomatic indices which are necessary and sufficient for the identification of the morbid conditions.

In the equation above P_{kj} are the a priori probabilities that the k^{th} disease Z_{kj} will occur in the population under study; X_j is the actual normalized average value of the index or symptom and W_{kj} is the cybernetic significance of the symptom X_j for a given disease Z_{kj}. The latter may be determined prior to this analysis by means of the method of Characteristic Intervals which I described in an earlier paper. The determination of these weights or coefficients constitutes some kind of a scanning process. A regression method may also be used for this purpose.

Dynamic relationships of this kind are given by:

$$Z_{ij} = P_{ij} \sum_{j=1}^{m} W_{ij} X_j(t) \qquad (2)$$

The memory of our system will consist of either magnetic tapes with analog data or, alternatively, electronic units for generating these relationships for each predefined disease of a given problem area S. The diagnosis will consist in comparing the incoming functions (signals) with the ones in memory.

Let us briefly discuss the general properties and organization of the system itself. In essence, it is a system which in each movement of time $t = 1, 2, \ldots, k$, receives a finite number of signals $x_1, x_2, \ldots, x_p$. Its internal behavior is governed by these signals or images and its possible responses are the possible diagnoses. These form a finite set $z_1, z_2, \ldots, z_n$. The responses are determined by the complex internal states of the machine; $\phi(z_1)$, $\phi(z_2)$, ..., $\phi(z_m)$ which constitue its memory capacity.

Static images or signals are analyzed according to the following scheme:

1) General processing,

2) Particular processing.

In general processing we evaluate the function $Z_{kj} = (x_1^k, x_2^k, \ldots x_m^k)$ for the incoming signals and compare this value to previously determined $Z_{min.}$ and $Z_{max.}$ for each problem area S until the optimal problem area for Z_{kj} has been determined (if this was not previously known). Once this has been done, then the gauge function $f_g(Z_{kj})$ for Z_{kj} is formed. The gauge function derives its existence from the following lemmas:

Lemma 1. If B is a convex body, Z_k any interior point, then any ray $Z+t(q-z_k), t \geq 0$ will cut the boundary S in one and only one point. Here Z is a point such that $Z = z_k + \phi(q-z_k)$, $\phi > o$.

Lemma 2. For any set of points in a convex body which satisfy Lemma 1, the gauge function f_g is given by

$(Z-z_k)f_g(q-z_k) = (q-z_k)$ where $q \neq z_k$ is any point of space and $z_k \in S$.

Therefore: $$f_g(q-z_k) = \frac{1}{\phi} = \lambda \qquad (3)$$

To show that this equation holds for all q in S we take the case where $q = z_k$ in which case $f_g(o) = o$. The gauge function is then defined for all points of space, and, in particular, for all points on the surface of the convex body S, the equation satisfied by all points of this surface::$f_g(q-z_k) = 1$. The inequality satisfied by all points of the minimal sphere containing S is $f_g(q-z_k) \leq 1$.

With gauge function formation particular processing of the composite signal begins. The next step is to perform spatial or spatio-temporal transformation (dispersion) of this function. Then correlational analysis is performed on it using the transformed functions in memory. In this manner the closest match is found. This match is the diagnosis.

For dynamic or random signals an alternative method for general processing exists. This consists in generating probability functions (continuous or discrete) $\mathrm{Prob}_s Z_k$ or $(1 - \mathrm{Prob}_s Z_k)$ that Z_k will belong to S or $\not{S}$ (not S), respectively, either by means of some type of a Monte Carlo method or by effecting the probability density function of the composite random variable Z_k. In the first case Z_k may be related to either set or problem area, in accord with these probabilities. In the second case, similarly, we assume that there is associated with each vector $Z_k(x_1^k, x_2^k, \ldots .. x_m^k)$ a probability distribution or density function and that the unknown vector is assigned to the category in whose region it falls on the basis of these probability functions.

In practice, the generated functions are sampled and averaged over a given time τ for a periodic signal and for a given number of cycles if the signal is periodic. The probabilities for the various problem areas are compared and the highest probability problem area S is selected. Particular processing here consists of transforming the probability functions and either autocorrelating or cross-correlating the transforms as indicated above.

When applying these methods to an analog system based upon the model constructed here, we must provide an exact definition of the signal or symptom in terms of phase, amplitude and frequency at the electrical level. In this case, as is well known, information content of such a signal is dependent upon the resolution limits for the detection of minimum as well as maximum amounts of energy representing the signal. In addition to this it is imperative that we determine the nature of the stochastic process we are dealing with - in particular whether it is stationary, non-stationary or homogeneous or ergodic (stationary and homogenous). In this way we can assume appropriate models in connection with processing the information.

III. A CLOSER VIEW OF THE SYSTEM

SOME OPTICAL ANALOGIES

Let me begin this section by drawing an analogy: Suppose that we regard each cybernetically significant index or symptom as a point source of monochromatic or quasimonochromatic coherent light signals. Suppose further that these signals have different frequency patterns (or intensities) and are emanating from the same subject. If in such a set-up the various signals are forced through one aperture, a diffracted interference pattern is formed. In the proposed system the interference pattern takes the form of a transformed composite pattern resulting from the convex combination of the signals. In this section we shall concern ouselves with the processing of the resultant pattern. But since in general the properties of the pattern, such as amplitude and frequency, are not well defined functions of time, we must describe them by statistical means such as averaging, and effecting spectral distributions as well as auto-correlations and cross-correlations.

As a consequence of the analogy drawn above, the following physico-mathematical considerations must be taken into account in designing the system:

1) A provision must be made at the input stage of the signal processor for a network to measure the magnitude of the signals received through each channel.
2) The coherence (i.e., phase constance) of the signals and the frequencies (if any) must be determined at the input stage so that the nature of the concomitant interference can be determined.
3) The diffracted interference pattern must be represented in our system by a spherical Fourier Transform (dispersion) since the individual signals must be defined within the unit sphere by virtue of the convexity criterion. (Note that the Fourier Transform is essentially the Wiener Amplitude Spectrum of a function.)
4) The transformed composite signal must be resolved through a matched and high pass filter.
5) The recongition stage must constitute a network for modulating the signal and generating correlation functions between the incoming signal, the signal stored in memory and a unit for comparing the values of these functions in order to determine the highest correlation.
6) These correlations must be displayed on a C.R.T. and at the same time digitized, then printed at the output stage.

To summarize these points, our system will be comprised of a) an input or scanning stage, b) a transforming stage, c) a modulation and recognition or decision stage, and d) an output stage. In the succeeding analysis, we shall concern ourselves principally with the stages b) and c).

We begin this analysis by noting that the gauge, probability and characteristic functions which we form from the incoming data are defined in spherical space - in particular, within the unit hypersphere.

This means that they are radial functions and that the transforms we effect from them are also radial functions as is established in harmonic analysis. In particular, the following general theorem from harmonic analysis applies to all radial functions in $L^1 (E_n)$ and $L^2 (E_n)$.

THEOREM: Suppose $n \geq 2$ and $f \in L^2 (E_n) \cap L^1 (E_n)$ has the form $f(Z) = f_o(/z/) P (Z)$, where $P(Z)$ in a solid spherical harmonic of degree k, then f has the form $\hat{f}(z) = f_o(/z/) p(z)$ where

$$F_o(R) = 2\pi i^{-k} R^{-\left[(n+2k-2)/2\right]} \int_o^\infty f_o(s) J_{(n+2k-2)/2} (2\pi rs)\, S^{(n+2k)/2} ds \quad (4)$$

This general identity describing the action of the F.T. on spherical space naturally leads to the investigation of the Bessel functions $J_m (R)$ which are involved in the integral. Near the origin we have the trivial estimate: $J_m (R) \sim CR^m$ as $R \rightarrow o$. The behavior for large R is less obvious and is given in the following lemma.

LEMMA: $J_m (R) = \sqrt{2/\pi R} \cos(R - m/2 - \frac{\pi}{4}) + O(R^{-\frac{3}{2}})$ as $R \rightarrow \infty$

In particular $J_m (R) = O (R^{-\frac{1}{2}})$ as $\rightarrow \infty$ (5)

Instead of using the continuous Fourier Transform one may also use the Bochner Transform method to implement the radial transformations proposed here. This method is described in some detail in my earlier paper quoted in the bibliography at the end of the present paper.

In order to implement the transformations digitally (in discrete fashion) one may make use of the Discrete Fourier Transform (DFT) of Cooley and Tuckey. Using this method the D.F.T. of a signal is defined as

$$Z(k\Omega) = \sum Z(nT) e^{-i\Omega Tnk}, \quad k=o, 1, \ldots, N-1. \quad (6)$$

In this equation $i = \sqrt{-1}$ and $\Omega = \frac{2\pi}{NT}$, where N is the number of signal samples to be transformed. In brief, the, the D.F.T. makes for a discrete representation of the continuous incoming signal. References for this method abound in the literature and the reader is referred to the bibliography at the end of this paper.

Once the composite signal has been defocused or dispersed, it may be channelled into a matched filter and then a high pass filter. This is done for three reasons: a) in order to effect selective data reduction, b) in order to enhance the relevant data, i.e., the Spectra, and reject or attenuate the lower frequency components, c) to optimize the signal-to-noise ratio. The filtered signals are now channelled into the recognition stage. Here one of two things happens: 1) We can either compare the incoming signals with those in memory by computing the sum of the differences (or the sum of the squares of the differences) between successive spectra or frequency lines. (The greater this quantity, the more patterns will be distinct.) 2) Or we perform cross-correlation analysis.

The basis of spectral analysis of signals is to be found in the generalization of Fourier and Fourier-type transformations and the fact that every structure Z_k has a finite energy Ω so that its integral:

$$\gamma = \int_{En} Z_k^2 \; dE \quad \text{exists and has a finite value.}$$

The pair-wise determination of the similarity between the two sets of signals (incoming and stored) proceeds by way of cross-correlation. In general we would be dealing with aperiodic and random signals. The relevant expressions in this case are:

$$\text{Aperiodic:} \quad R_{zy}(\tau) = \int_{-\infty}^{\infty} Z(t)\; Y(t-z)\; dt$$

$$\text{Random:} \quad R_{zy}(\tau) = \frac{1}{T-\tau}\int_{0}^{T-\tau} Z(t)\; Y(t-\tau)\; dt \qquad (7)$$

These expressions may be evaluated either digitally or continuously. Thus the digital implementation of cross-correlation for aperiodic signals can be approximated by a sampling method:

$$R_{zy}(\tau) = \frac{1}{N} \sum_{k=1}^{N} Z(k\Delta t-\tau)\; Y(k\Delta t). \qquad (8)$$

Finally, the decision or diagnosis is made by a comparator (either digital or analog). This unit selects the signal in memory which effects the highest correlation with the incoming signal as the diagnosis. This diagnosis may be checked by means of further tests on the patient and if confirmed, the new signal may at a later time be made available to modify the memory in order to improve the accuracy of future diagnoses. If the diagnosis is not supported by the tests then different problem areas must be tried. The final as well as intermediate results of each analysis must be displayed on a C.R.T. and printed at the output stage.

IV. TERMINAL REMARKS

In this paper I have dealt principally with the preliminary design of a cybernetic data processor and simulator based on a mathematical conceptual model. For this reason I have dwelt at some length on the mathematical basis of the expected behavior of such a system in a particular context --- medical differential diagnosis.

The aim of cybernetics is to delve at the mechanisms, laws and phenomena connected with control, information processing transmission, storage and retrieval in complex man-made systems as well as in organic (living) systems which exist in nature. It is worth noting that it may or may not be possible to reduce the general laws or mechanisms of living system behavior to those of a machine. For it is plain that complex principles and mechanisms are manifest in living systems as have not yet or perhaps cannot be simulated on machines.

The living brain constitutes a highly complex system of interconnected nerve elements. Individual nerves possess relatively simple properties. They can either either be in a state of excitation or in a state of inertia. They may either conduct an excitation wave or fail to conduct it --- according to the all-or-none law. The remarkable properties possessed by the brain (thought, imagination, memory,...etc...) occur only in highly complex specially interconnected elements. Thus a study of separate nervous elements or separate groups of them, no matter how precise the method by which it is pursued is not likely bring us to an understanding of the fundamental working principles of the brain or of those faculties it possesses. As far as the brain is concerned, the whole is greater than the sum of the functions of the parts.

We know very little about the operation of the brain and the neuro-physiological structure of its cortex. It is clear from the above, then, that great importance must be attached to the methods for simulating the processes taking place in physiological systems. In this connection, the brain can be imagined as a complex cybernetic system which we cannot watch being built or constructed. By simulating the individual functions of this organ using cybernetic systems, we can form various hypothesis as to its structure.

In an age of brain measurements that fail to yield models for what the brain does, and of unstructured cybernetic networks, that are expected, when properly adjusted to perform wonders, it is expedient that we seek hypotheses at an appropriate level of organization and use interdisciplinary approaches or analyses to solve cybernetic problems. One advantage of the interdisciplinary approach is that phenomena in one field of investigation may be recognized in another field of study and it is worth the effort to consider adapting the techniques and methodology of the first field to the second field.

For this reason cyberneticists must make it their business to acquire expert or near-expert knowledge in many diverse sciences such as biology, physics, psychology, computer science, psycho-physics, electronics and the philosophy of science, to mention but a few. Because natural cybernetic systems possess facets that may be interpreted through these and other sciences it may be speculated that the substance out of which the mind-brain is made is the ultimate substance out of which the universe is constructed. If this is the case, then the discovery of the fundamental secrets of the mind-brain may have to await further development or new formulations in quantum physics, quantum mechanics, quantum chemistry and quantum electronics.

But if, as is most likely to be the case, according to this author, man cannot comprehend or, for that matter, apprehend (control) this ultimate or absolute reality in its unmitigated purity no matter what tools he uses, the best he can do is to approach it constantly; coming closer to the truth, as it were, but never quite reaching or realizing it!

LITERATURE

1) Arbib, Michael A., Theories of Abstract Automata, Prentice-Hall, Englewood Cliffs, New York, 1969.

2) Banda, J. M., A Mathematical Model for Computer Simulation of the Process of Differential Diagnosis in Medicine Proceedings: Third Annual Pittsburg Conference on Modelling and Simulation, University of Pittsburg, Pennsylvania, 1972.

3) Carcers, Caesar A., and Rikli, Arthur E., Diagnostic Computers, Charles C. Thomas Inc., Springfield, Illinois, 1969.

4) Collier, R. J., Burckhardt, C. B. and Lin, L. H., Optical Holography, Academic Press, New York, 1971.

5) Gold, Bernard and Rader, Charles M., Digital Processing of Signals, McGraw-Hill Book Company, New York, 1969.

6) Wilson, Ira G. and Wilson, Martham,E., From Idea to Working Model, John Wiley and Sons, New York, 1970.

7) Stein, M. Elies and Weiss, Guido, Introduction to Fourier Analysis on Encledian Spaces, Princeton University Press; Princeton, 1971.

A DYNAMIC LABOR ALLOCATION MODEL INCLUDING EFFICIENCY

MARLIN H. MICKLE and WILLIAM G. VOGT
University of Pittsburgh
Pittsburgh, Pennsylvania

ABSTRACT

The use of dynamic economic models based on the work of Leontief has received increased attention due to activity in the dynamic analysis of urban and world areas. Stable dynamic models have been developed and consideration of labor in these areas has been proposed. The optimum allocation of labor accounting for varying efficiency is considered in the dynamic model presented in this paper. The availability of labor as a sociological output is introduced.

INTRODUCTION

An economic system can be described by a table or matrix commonly **termed** transactions table or matrix where each row and column represents a particular grouping or aggregation of industries. The input-output models of an economic system based on this table were developed by Leontief.(5)

Strictly speaking, one industry may produce more than one commodity. However, each commodity is linked with its producing portion of industry in this paper. Thus commodity j is assumed to be produced by industry j without any loss to generality and will be so identified. Industry j is therefore that portion and all associated production capability devoted to commodity j and may be part of a larger corporate structure producing other commodities so identified although not affecting present considerations.

The models discussed in this paper account for two types of outputs. The first is an output which is used by an industry as an intermediate product or material in the production of some other product. The second is an output which is available to satisfy a demand outside the producing industries. The inter-industry portion of the model can be represented by an nxn matrix, $\underline{U}$, where u_{ij} is the output of industry i input to industry j. An nxm matrix, $\underline{E}$, where e_{ij} is the output of industry i going to satisfy external demand j then represents the external demand portion giving the complete transactions matrix, $(\underline{U},\underline{E})$.

The static model of such an economic system contains the following basic variables. The rate of production of the i-th commodity, x_i; the rate at which the i-th commodity is used to produce one unit of the j-th commodity per unit time, a_{ij}; the supply rate of commodity i available for external consumption, s_i; the external demand rate for commodity i, d_i; b_{ij} is the stock of commodity i associated with increase of level or activity producing commodity j; and the matrix giving the dynamics of the individual processes, $\underline{F} = (f_{ij})$.

The static model of Leontief is given by

$$\underline{x} = \underline{A}\,\underline{x} + \underline{s} \tag{1}$$

$$(\underline{I}-\underline{A})\underline{x} = \underline{s} \tag{2}$$

The matrix $(\underline{I}-\underline{A})$ is termed the Leontief matrix if the matrix $\underline{A}$ is viable.(3)

THE DYNAMIC MODEL

A typical analysis using the static model is based on an open type model which supplies the indicated external demands where the labor required in the production process is not taken into account. Subsequent work attempted to include dynamics in the model of Leontief. However, these attempts have resulted in models which are unstable.(2,3,4)

The authors have developed stable models which take into account the dynamics of the economic system(1) and labor.(6) Typical earlier models were of the dynamic equilibrium type where the supply is said to increase or decrease to satisfy an increase or decrease in demand. The dynamic models developed by the authors allow both supply and demand to vary independently with the difference acting as an error to provide the drive for the system.

The resulting dynamic models, excluding labor considerations, are given below with the previously defined variables. The Type I model is given by equations (3) and (4), and the Type II model is given by equations (5) and (6).

MODEL I

$$\dot{\underline{x}} = -F[\underline{I}-\underline{A}(\underline{I}-\underline{A}+\underline{BFA})^{-1}\,(\underline{I}-\underline{A}+\underline{BF})]\underline{x} + \underline{F}[\underline{I}-\underline{A}(\underline{I}-\underline{A}+\underline{BFA})^{-1}\underline{BF}]\underline{d} \tag{3}$$

$$\underline{s} = (\underline{I}-\underline{A})(\underline{I}-\underline{A}+\underline{BFA})^{-1}\,(\underline{I}-\underline{A}+\underline{BF})\underline{x} - (\underline{I}-\underline{A})\,(\underline{I}-\underline{A}+\underline{BFA})^{-1}\underline{BFd} \tag{4}$$

MODEL II

$$\dot{\underline{x}} = -\underline{F}(\underline{I}-\underline{BF})^{-1}\,(\underline{I}-\underline{A})\underline{x} + \underline{F}(\underline{I}-\underline{BF})^{-1}\underline{d} \tag{5}$$

$$\underline{s} = (\underline{I}-\underline{BF})^{-1}(\underline{I}-\underline{A})\underline{x}-\underline{BF}(\underline{I}-\underline{BF})^{-1}\underline{d} \tag{6}$$

LABOR CONSIDERATIONS

As mentioned previously, the initial developments of both the static and dynamic models are open models in that they do not consider labor. The initial consideration of labor in the dynamic model is presented with a total of M workers with p_i workers in industry i; productivity, γ_{ii} (goods/man-hour); and length of work day, τ_{ii} (work-hours/day). Based on these variables, a labor dominated dynamic process is given by

$$\dot{\underline{x}} = \dot{\underline{\gamma}}\,\underline{\tau}\,\underline{p} + \underline{\gamma}\,\dot{\underline{\tau}}\,\underline{p} + \underline{\gamma}\,\underline{\tau}\,\dot{\underline{p}} \tag{7}$$

where with $\dot{\underline{\gamma}} = \dot{\underline{\tau}} = \underline{0}$, $\dot{\underline{x}} = \underline{\gamma}\,\underline{\tau}\,\dot{\underline{p}}$. Now neglecting stock considerations, $\underline{B}=\underline{0}$, equation (3) becomes

$$\dot{\underline{x}} = \underline{\gamma}\,\underline{\tau}\,\dot{\underline{p}} = \underline{-F}(\underline{I}-\underline{A})\underline{x} + \underline{F}\;\underline{d} \tag{8}$$

which implies

$$\underline{\gamma}\,\underline{\tau}\,\underline{w} = \underline{F}(\underline{d}-\underline{s}) \tag{9}$$

where $\underline{w}$ is a decision function. Thus a reasonable linear model is given by

$$\underline{w}(t) = \underline{R}[\underline{d}(t)-\underline{s}(t)] \tag{10}$$

where $\underline{R}$ is a decision gain matrix. Therefore,

$$\underline{F} = \underline{\gamma}\ \underline{\tau}\ \underline{R} \tag{11}$$

In the previous work,$^{(6)}$ the cases of Excess Labor (High Unemployment) and Insufficient Labor (Low Unemployment) were considered with the Insufficient Labor case providing an interesting problem which was formulated as a linear programming problem. The number of worker categories or occupations is finite and is considered as m.

Now consider the problem with the above m occupations and in addition, n industries where some subset of the m occupations is required by industry j. This is the point from which the allocation problem is formulated in this paper.

THE OBJECTIVE FUNCTION

The primary objective function of the previous work$^{(6)}$ is a linear combination of the elements of $\underline{F}$ in model equation (3) which is to be maximized in order to minimize the response time of the dynamic system. Each element of $\underline{F}$, f_{ij}, is a function of $\underline{\gamma}$, $\underline{\tau}$ and $\underline{R}$; i.e., $f_{ij} = (\underline{\gamma}\ \underline{\tau}\ \underline{R})_{ij}$ which can be observed from equation (11). A previous assumption$^{(6)}$ by which any worker in occupation i can function equally well in any industry of the corresponding subset is not realistic and will be relaxed by the incorporation of efficiency. Thus the efficiency of any category will no longer be assumed to be one and will be free to vary over a continuous range. Now the efficiency of a worker of occupation i performing in industry j is given by α_{ij} with

$$0 < \alpha_{ij} \leq 1 \tag{12}$$

where a small value of α_{ij} indicates little efficiency i.e., the worker will have great difficulty with the job, and 1 indicates 100% efficiency.

It is important to note that the controversial situation of making the fundamental statement that there exist jobs which a person or class of persons can not do at all, i.e. $\alpha_{ij} = 0$ has been avoided. This point is not a loss to generality in an area where quantification is difficult at best because extremely small values of α_{ij} can be used to produce the same result, accounting for measuring inaccuracies, without introducing an argumentative point.

The previously obtained objective function$^{(6)}$ for this type of system incorporation labor is given by

$$J = \sum_{j=1}^{n} \sum_{i=1}^{n} f_{ij} = \sum_{j=1}^{n} \sum_{i=1}^{n} (\underline{\gamma}\ \underline{\tau}\ \underline{R})_{ij} \tag{13}$$

where for fixed $\underline{\gamma}$ and $\underline{\tau}$, J is to be maximized. For the remainder of this work, it will be assumed that the $\underline{F}$ and $\underline{R}$ matrices are diagonal resulting in the following objective function:

$$J = \sum_{j=1}^{n} f_{jj} = \sum_{j=1}^{n} (\underline{\gamma}\ \underline{\tau}\ \underline{R})_{jj} \tag{14}$$

The constraints on the previous optimization problem$^{(6)}$ were obtained from equation (10) as shown in (15) where $h_i(t)$ is the available rate of persons of occupation i at time t.

$$w_i(t) \le h_i(t) : i = 1,2,\ldots,m \tag{15}$$

Inequalities of the following type can then be obtained:

$$\sum_{j=1}^{n} r_{jj}[d_j(t)-s_j(t)] \le h_i(t) : i = 1,2,\ldots,m \tag{16}$$

In addition, the following constraints also apply:

$$r_{ii} \ge 0 : i = 1,2,\ldots,n \tag{17}$$

The solution of the problem of maximizing (14) subject to (16) and (17) can be solved directly using linear programming.(6)

Now consider the objective function given by (14) applying the following relationship:

$$r_{jj} = \sum_{i=1}^{m} \alpha_{ij}\, y_{ij} : j = 1,2,\ldots,n \tag{18}$$

where y_{ij} is a gain (workers/goods) where the workers have occupation i and are assigned to industry j. α_{ij} is the previously defined efficiency. The resulting objective function, applying (18), is given by

$$J = \sum_{j=1}^{n} \gamma_{jj}\, \tau_{jj} \sum_{i=1}^{n} \alpha_{ij}\, y_{ij} \tag{19}$$

$$J = \sum_{j=1}^{n} \sum_{i=1}^{m} \gamma_{jj}\, \tau_{jj}\, \alpha_{ij}\, y_{ij} = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}\, y_{ij} \tag{20}$$

where

$$c_{ij} = \gamma_{jj}\, \tau_{jj}\, \alpha_{ij} : i = 1,2,\ldots,m;\ j = 1,2,\ldots,n \tag{21}$$

CONSTRAINTS

The system error produced by industry j is given by $[d_j(t)-s_j(t)]$. The rate of available workers in the i-th occupation at time t is given by $h_i(t)$. Now one of the decisions which must be made is the allocation of workers to various industries where the rate at which the workers are assigned is a function of the demand for goods, i.e. the workers are required according to demand.

The basic objective of the optimization problem posed is the ability of industry as a whole to respond to an error in minimum time. The objective is not primarily one of monetary considerations. Each element of the error represents one item or group of items which must in turn be allocated among different occupations or different industries. Two problems are posed. The first is what ratio of each error element is allocated to workers of each of the m occupations. The second is the rate (workers/goods) or gain assigned to industry j. Thus with m occupations and n industries, $\underline{\beta}$, is an m by n matrix where the indicated portion of error element j assigned to occupation i is given by β_{ij}. $\underline{\beta}$ is a type of decision matrix dictated by particular industries and represents policy or policies arrived at indepently as part of a seperate problem. Thus with $h_i(t)$ the rate of available workers in occupation i, the following constraints are obtained:

$$\sum_{j=1}^{n} y_{ij} \beta_{ij} [d_j(t)-s_j(t)] \leq h_i(t): i=1,2,\ldots,m \tag{22}$$

Obviously each industry, j, can not accept an unlimited number of workers due to limitations such as equipment, space, etc. The maximum rate at which workers can be taken into industry j at time t is given by $v_j(t)$. Thus, the following constraints are obtained:

$$\sum_{i=1}^{m} y_{ij} [d_j(t)-s_j(t)] \leq v_j(t) : j=1,2,\ldots,n \tag{23}$$

Obviously, the following constraints are imposed on the matrix $\underline{\beta}$:

$$\sum_{i=1}^{m} \beta_{ij} = 1 : j=1,2,\ldots,n \tag{24}$$

The following constraints on the y_{ij} elements apply:

$$y_{ij} \geq 0 : i=1,2,\ldots,m; j=1,2,\ldots,n \tag{25}$$

The problem which has just been formulated can now be summarized. The objective function which is to be maximized is given in (26) where the variables y_{ij} are subject to the constraints (27), (28) and (29).

$$J = \sum_{j=1}^{n} \sum_{i=1}^{m} \gamma_{jj} \tau_{jj} \alpha_{ij} y_{ij} \tag{26}$$

$$\sum_{j=1}^{n} y_{ij} \beta_{ij} [d_j(t)-s_j(t)] \leq h_i(t): i=1,2,\ldots,m \tag{27}$$

$$\sum_{i=1}^{m} y_{ij} [d_j(t)-s_j(t)] \leq v_j(t) : j=1,2,\ldots,n \tag{28}$$

$$y_{ij} \geq 0 : i=1,2,\ldots,m ; j=1,2,\ldots,n \tag{29}$$

The above optimization problem can obviously be solved using linear programming.

RATE CONSIDERATIONS

In constraints (27) and (28), the upper bounds are rates, i.e. people/day. $v_j(t)$ is the maximum rate at which industry j can accept workers, and $h_i(t)$ is the rate at which people are available in occupation i. The $v_j(t)$ is determined by the jth industry and for the purposes of this discussion will be assumed to be determined strictly by the industry. However, it is recognized that various parameters in the industry may be adjusted to obtain an alternate $v_j(t)$ which would result in a larger value of J in (26).

The rate $h_i(t)$ is of particular interest at this point. The rate at which people are available in any occupation is a function of many factors such as parent's occupation, education, perceived values, social mobility in the society in question, etc. Thus the rate $h_i(t)$ poses an interesting problem. Sociologists have developed what is termed stratification theory which deals with the situation of the various occupations or worker categories mentioned above. The ability to move up or down in this stratification can be described by social mobility where the child of a person in occupation i can move to occupation j with mobility m_{ij} with $0 \leq m_{ij} \leq 1$ where $(m_{ij}) = M$ is an mxm matrix termed the mobility matrix.

The social system involving stratification and mobility is a dynamic system which can be modeled(8) in a manner very similar to that of the dynamic Leontief model. The rate at which workers are available to supply an external demand in the dynamic sociological model(8) is given by $\underline{h}$ with each element h_i being the h_i of the previous discussion (constraint 27). With appropriate definitions of the $\underline{F}$ and $\underline{A}$ matrices and $\underline{y}$ being the birth rate, where y_i is the rate at which children are born to persons in the ith occupation, the following dynamic sociological model is obtained:(8)

$$\dot{\underline{y}} = -\underline{F}(\underline{I}-\underline{A})\underline{M}'\,\underline{y} + \underline{F}\,\underline{d} \tag{30}$$

$$\underline{h} = (\underline{I}-\underline{A})\underline{M}'\underline{y} \tag{31}$$

The rate at which workers are assigned to industry j has previously been indicated by $w_i(t)$.(6) In addition, it has been shown that

$$\underline{p}(t) = \underline{p}(o) + \int_{o}^{t} \underline{w}(\alpha)d\alpha \tag{32}$$

where $p_i(t)$ is the number of workers in industry i with

$$w_i(t) \leq h_i(t) \tag{33}$$

and

$$\dot{\underline{x}}(t) = \underline{\gamma}\;\underline{\tau}\;\dot{\underline{p}}(t) = \underline{\gamma}\;\underline{\tau}\;\underline{w}(t) \tag{34}$$

Thus the problem of allocation of labor as posed in (26) through (29) is not only a static optimization problem at any particular point in time. In addition, it provides a link or interface between a dynamic economic model(1) and a dynamic sociological model.(8)

POLICIES

Many policies can be proposed and evaluated using the dynamic socioeconomic system just developed. However, the space and topic of this paper limit the many alternatives and only one example is presented. Specifically, consider the policy of full employment in which the constraints (27) now become equalities rather than inequalities. In addition, assume this rate of labor over all occupations is such that it is exactly sufficient to satisfy the maximum rates at which industry can accept workers. Thus the constraints (28) also become equalities.

Substituting equalities in (27) and (28) produces an allocation problem in which the coefficients of the variable y_{ij}'s are not unity as in the classical transportation problem. However, the problem posed of maximizing (26) subject to (27) through (29) with equalities replacing inequalitiies is the weighted distribution problem.(7)

SUMMARY

The results of this paper are two-fold. First, the problem of obtaining an optimum response in a dynamic economic system is formulated as an allocation problem in which the availability and efficiency of persons are taken into account. This result is formulated as a linear programming problem and, after allowing a particular policy, is formulated as a weighted distribution problem. Second, the composition of the above result establishes an interface between a dynamic economic and a dynamic sociological model providing one form of a dynamic socio-economic system.

REFERENCES

1. Vogt, W.G., Mickle, M.H., and Aldermeshian, H., "A Dynamic Input-Output Model for an Economic System," Proceedings of the Ninth Allerton Conference on Circuit and System Theory, 1971.
2. Sarjan, J.D., "The Instability of the Leontief Dynamic Model," Econometrica, 26 (1958) 381-392.
3. Chakravarty, S., Capital and Development Planning, MIT Press, 1969.
4. Hawkins, D., and Simon, H.A., "Note: Some Conditions of Macro-Economic Stability," Econometrica, 17 (1949) 245.
5. Leontief, W.W., The Structure of the American Economy 1919-1939, Oxford Press, 1951.
6. Vogt, W.G. and Mickle, M.H., "Labor in a Dynamic Economic Model," Proceedings of the Sixth Princeton Conference on Information Sciences and Systems, 1972.
7. Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, 1963, Chapter 21.
8. Mickle, M.H. and Vogt, W.G., "A Dynamic Sociological Model," to be published.

A STOCHASTIC MODEL FOR A DYNAMIC LEONTIEF SYSTEM

WILLIAM G. VOGT and MARLIN H. MICKLE
University of Pittsburgh
Pittsburgh, Pennsylvania

ABSTRACT

There are many techniques and concepts from control theory which are of direct use in the modeling of economic systems. When properly matched with classical economic modeling, these techniques can lead to new insights about the operation of complicated economic processes and can lead to new and important results.

INTRODUCTION

One description of an economic system is given by a table or matrix giving the inputs and outputs from various industries. The two types of outputs are the output to other industry which is used by the industry to produce an output, and the output which is available to satisfy an external or final demand. The inter-industry portion is a square matrix $\underline{V}$ with the final demand being given by a matrix $\underline{D}$. With n industries and m types of final demand, $\underline{V}$ is n by n, and $\underline{D}$ is n by m. This is essentially a static description.

The n by n+m matrix $(\underline{V},\underline{D})$ is commonly called an input-output table or transactions table. Many industries exhibit similar characteristics making it unnecessary to account for the n individual industries. It is possible to aggregate various industries, thus reducing the dimensions of the original matrix or transactions table. If after the aggregation, the n industries are reduced to p industries, $\underline{V}$ is aggregated(1) into a p by p matrix, and $\underline{D}$ is aggregated into a pxm matrix, i.e. $(\underline{V},\underline{D})$ is transformed into the matrix $(\underline{U},\underline{E})$ having dimension p by p+m.

Many attempts at accounting for or including dynamics in such a model resulted in an unstable model.(2,3,4) A recent development in the modeling effort is a model which includes the dynamic aspects of the system with a stable model.(1) In addition, the recent model in the static case is consistent with Leontief input-output theory. However, while accounting for system dynamics, the newly developed model has been described in an entirely deterministic way.

The purpose of the present research reported in this paper is the extension of this model to account for stochastic inputs resulting from measurements and the non-deterministic nature of some of the variables in the model.

THE STATIC MODEL

The following system variables are defined for i, j=1,2,...,n:

x_i = the rate of production of the i-th commodity;

a_{ij} = the rate at which the i-th commodity is used to produce one unit of the j-th commodity per unit time;

b_{ij} = the stock of commodity i associated with increase of level of activity producing commodity j;

s_i = the supply rate of commodity i available for external consumption;

d_i = the external demand rate for commodity i;

f_{ij} = the matrix element giving the dynamics of the process.

$\underline{x}$, $\underline{s}$ and $\underline{d}$ are matrices having dimension nxl, and $\underline{A}$, $\underline{B}$, $\underline{F}$ are matrices having dimension nxn.

The static matrix model is given by

$$\underline{x} = \underline{A}\ \underline{x} + \underline{s} \tag{1}$$

or, rearranging

$$(\underline{I}-\underline{A})\ \underline{x} = \underline{s} \tag{2}$$

The matrix $(\underline{I}-\underline{A})$ is known as the Leontief matrix if the matrix $\underline{A}$ is *viable*,(3) and the static system given by (2) is known as the static Leontief model.

THE DYNAMIC MODEL

Two important variables in any economic system are supply and demand. Many dynamic models do not provide for both of these variables to vary separately. Instead, the supply is said to increase to satisfy demand, or the supply will decrease to a reduced demand. Models of this type are said to be in *dynamic equilibrium*. The recently developed model allows for both of these variables to vary separately with the difference providing the "drive" for the system.

As indicated in previous papers,(1,5) there are two variations on the dynamic model, Type I and Type II.

For the sake of illustration, only the Type II model will be considered here. The equations of the Type II model are given in (3) and (4), which are obtained from the signal flow diagram of Figure 1.

$$\underline{\dot{x}} = -\underline{F}(\underline{I}-\underline{BF})^{-1}\ (\underline{I}-\underline{A})\underline{x} + \underline{F}(\underline{I}-\underline{BF})^{-1}\underline{d} \tag{3}$$

$$\underline{x} = (\underline{I}-\underline{BF})^{-1}\ (\underline{I}-\underline{A})\underline{x} - \underline{BF}(\underline{I}-\underline{BF})^{-1}\underline{d} \tag{4}$$

If $\underline{B} = \underline{0}$, (3) and (4) reduce to

$$\underline{\dot{x}} = -\underline{F}(\underline{I}-\underline{A})\underline{x} + \underline{F}\ \underline{d} \tag{5}$$

$$\underline{s} = (\underline{I}-\underline{A})\underline{x} \tag{6}$$

THE ERROR DETECTOR

In the formulation above it was assumed that the demand vector, $\underline{d}$, was a known deterministic quantity. In case it is, and assuming knowledge of the state of the system at some time, then the output of the system, the supply vector, $\underline{s}$, becomes a calculable function of time.

However, the actual structure of an industrial system in the vicinty of the node marked $\underline{e}$ in Figure 1 would not be as indicated. Instead, it would consist of a block which processes information made available to it and comes up with a decision about whether to increase or decrease production. This is shown schematically in Figure 2. Insofar as this block is concerned, a linearized version would still yield the signal flow diagram of Figure 1.

The information supplied to the block of Figure 2 includes the demand vector, $\underline{d}$, and the supply vector, $\underline{s}$, both assumed to be deterministic. If we examine a little more carefully the concept of demand, it becomes clear that a stochastic model is more akin to what actually takes place.

THE DEMAND VECTOR

Demand for a good exists if any can be sold. If demand does not exist, inexorably the good will go out of production. Internal demand for a good can be measured in a more or less direct way. However, external demand, that which makes the industrial productive system work, is a little more ephemeral.

Generally speaking, the demand vector, $\underline{d}$, is not a physical variable that can be measured and quantified. What can be done is data collected regarding the number of a good sold, the number of orders made, the time of year of the sales, the segments of the population buying, etc. In short, information regarding the results of the demand for a good can be collected, but the demand itself remains so far unmeasured.

External demand, consumer demand, is probably the result of very complex and interacting social and mental processes. If this is the case, the desirable objective of direct measurement of this variable is still a long way off. Instead, we must be satisfied with indirect, and therefore imprecise, estimates of what the demand vector is.

The analysis of demand for a good is not unknown. There are many methods of approaching the problem of demand analysis, among them the general area of time series analysis,(6) the tools for which are largely statistics based on data which has been collected for this purpose.

As an example, one version of a time series analysis is to estimate the demand as a polynomial in time

$$\hat{d} = a_0 + a_1 t + \ldots + a_n t^n \qquad (7)$$

and use previoulsy collected data to identify the parameters $a_0, a_1, \ldots, a_n$. Thus, one has an identification problem, in which, generally speaking, one selects the parameters $a_0, a_1, \ldots, a_n$ to minimize a performance index of some type, which usually involves the error or scatter of the data around the estimated demand, $\hat{d}$.

Another version of a model is a recursive one

$$\hat{\underline{d}}(k+1) = \underline{a}(k)\, \hat{\underline{d}}(k) + \underline{b}(k)\underline{q}(k) + \underline{u}(k) \tag{8}$$

In this case, past data, $\{\underline{q}(k)\}$ is used to identify the matrices $\underline{a}(k)$ and $\underline{b}(k)$ so that the fit for the data collected gives certain statistical properties to the error $\underline{u}(k)$. Once again, the form and structure of the model is assumed, data is collected, identification is done and a demand model is obtained. In this case, means are provided to predict ahead based on certain current measurements, the value of $\underline{q}(k)$.

The obtaining of the demand model in either case follows that illustrated in Figure 3. Once the demand model is identified, it can be represented as in Figure 4. Figure 4(a) is a general case which could be expanded to include filtering if the dynamics of the social and mental processes ever become known. Figure 4(b) illustrates the more likely case where $\hat{\underline{u}}$ is viewed as measurement error of demand $\underline{d}$. This error would have certain statistical properties which might be related to the noise, $\underline{u}(k)$ of equation (8).

THE SUPPLY VECTOR

The case for the supply vector is similar to the case of the demand vector, except that more accurate information is available if a quality management information system is used. However, there will always be some noise introduced into this system through unexplained losses, etc. Thus, the information used to make decisions will not be the actual supply, $\underline{s}$, but the measured supply, $\hat{\underline{s}}$, which is related to $\underline{s}$ by

$$\hat{\underline{s}} = \underline{s} + \underline{v} \tag{9}$$

where $\underline{v}$ is the measurement error.

THE STOCHASTIC LEONTIEF MODEL

From the above developments, a stochastic version of a Leontief Model is direct and is illustrated in signal flow form in Figure 5. The equations for the stochastic system are (assuming $\underline{B} \equiv \underline{0}$):

$$\dot{\underline{x}} = \underline{F}\ \underline{e} \tag{10}$$

$$\underline{e} = \hat{\underline{d}} - \hat{\underline{s}} \tag{11}$$

$$\hat{\underline{d}} = \underline{d} + \hat{\underline{u}} \tag{12}$$

$$\hat{\underline{s}} = \underline{s} + \underline{v} \tag{13}$$

$$\underline{s} = (\underline{I}-\underline{A})\underline{x} \tag{14}$$

where $\hat{\underline{u}}$ and $\underline{v}$ are zero mean random vectors. At this stage of the research, nothing else can be said about the statistical descriptions of $\hat{\underline{u}}$ and $\underline{v}$.

CONCLUSIONS

A stochastic model for a dynamic Leontief system has been introduced based largely on the results of demand analysis. If the usual assumptions regarding the noise vectors can be made, the system closely approximates that of stochastic control theory.

REFERENCES

1. Vogt, W.G., Mickle, M.H., and Aldermeshian, H., "A Dynamic Input-Output Model for an Economic System," *Proceedings of the Ninth Allerton Conference on Circuit and System Theory*, 1971.

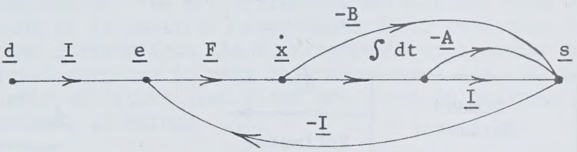
2. Sarjan, J.D., "The Instability of the Leontief Dynamic Model," *Econometrica*, 26 (1958) 381-392.

3. Chakravarty, S., *Capital and Development Planning*, MIT Press, 1969.

4. Hawkins, D., and Simon, H.A., "Note: Some Conditions of Macro-Economic Stability," *Econometrica*, 17 (1949) 245.

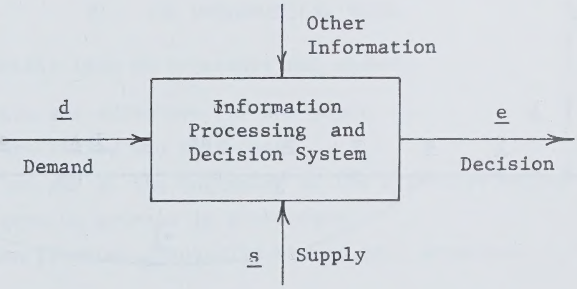
5. Vogt, W.G. and M.H. Mickle, "Labor in a Dynamic Economic Model," *Proceedings of the Sixth Princeton Conference on Information Sciences and Systems*, 1972.

6. Box, G.E.P. and G.M. Jenkins, *Time Series, Forecasting and Control,* Holden-Day, San Francisco, 1970.



-B
d
I
e
F
ẋ
∫ dt
-A
s
I
-I

Figure 1. Signal Flow Diagram of the Type II Model.



Other Information
d
Demand
Information Processing and Decision System
e
Decision
s
Supply

Figure 2. More detailed description of how e is obtained.

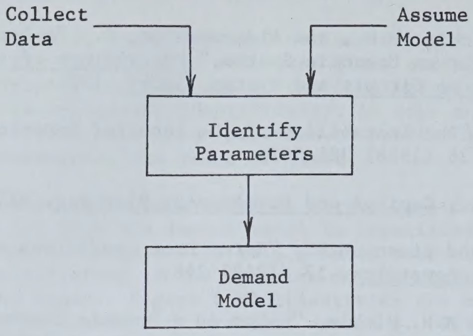
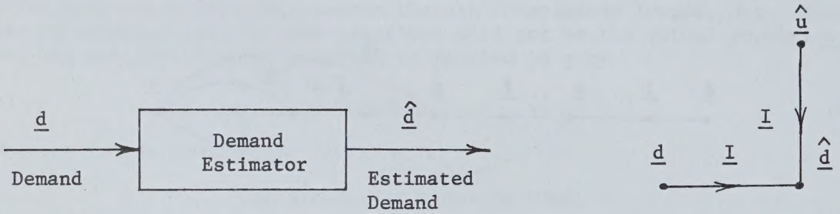

Collect Data
Assume Model
Identify Parameters
Demand Model

Figure 3. Obtaining a Demand Model.


d
Demand
Demand Estimator
d̂
Estimated Demand
û
I
d
I
d̂

(a)

(b)

Figure 4. Estimation of Demand: (a) General Case; (b) Linearized version.

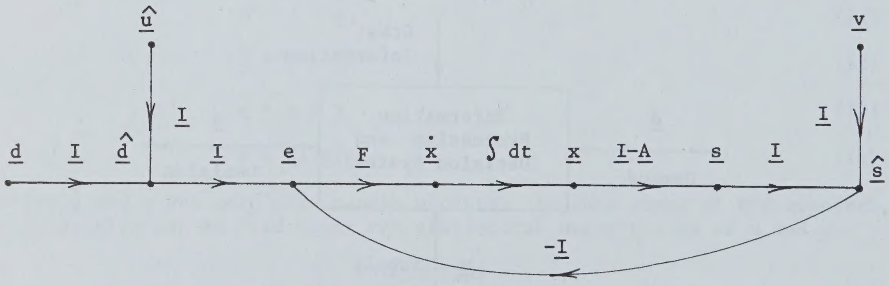

û
I
d
I
d̂
I
e
F
ẋ
∫ dt
x
I-A
s
I
v
I
ŝ
-I

Figure 5. Linearized Version of a Stochastic Model for a Dynamic Leontief System (It is assumed that $\underline{B} = \underline{0}$).

APPLICATION OF DISCRETE-DATA SYSTEM TECHNIQUES IN A FINANCIAL PROBLEM OF INSURANCE

JOHN W. CARON* TAPAN S. ROY** DANIEL TABAK***
The Travelers Insurance Co., Hartford, Conn. 06115

ABSTRACT The techniques of discrete-data systems analysis are applied to the modeling of a financial problem occurring in the insurance industry. In order to estimate unknown parameters, an unconstrained nonlinear programming problem is formulated and solved using the Nelder and Mead method.

I. INTRODUCTION

There are several types of cash transactions involved in the insurance business, such as payment of: premium commissions, state premium taxes, claim losses and receipt of premiums, which can be modeled as discrete-data systems. A particular problem, dealt with in this paper, is the evaluation of the paid premium component of an insurance company's cash flow forecast.

Insurance companies typically experience an inflow of paid premiums together with cash outflows for: claim settlements, commissions and various other expenses. The net difference between the cash inflow and outflow (assuming it is positive) represents funds available for investment. The income derived from these investments constitutes an important component of total earnings in many insurance companies. Consequently, accurate forecasts of future cash flows are vital in an insurance company's investment planning.

The premium paid is one of the key components of cash flow. The evaluation of parameters which influence this component is the main subject of this paper. A mathematical model (in the form of a discrete time system) of the premium paid subsystem of the cash flow is developed in section II. An unconstrained minimization scheme for the identification of the unknown parameters of the model is presented in section III with computational examples in IV. Concluding remarks are given in section V.

II. THE MATHEMATICAL MODEL

Consider a specific line of insurance and assume:

(1) All policies are effective for one year.

(2) Premiums are billed and paid once.

(3) Premiums are due at the beginning of the effective policy year.

(4) The paid premium process is stationary.

(5) All written premiums eventually become paid premiums.

---

* Research Associate, Corporate Research, Travelers Insurance Company
** Assistant Director, Corporate Research, Travelers Insurance Company
*** Travelers Research Fellow, Summer 1972. Dept. of Electrical Engineering, Univ. of Negev, Beer-Sheva, Israel, on leave from RPI of Connecticut, Hartford, Conn. 06120.

Let WP_k be the premium written during month k, and P_{kj} the paid premiums received during month j, which were written during month k. Assuming the paid premium process is stationary, we can express the probability of time of payment j of premiums, written at time k, as p(j-k).

$$p(j-k) = P_{kj} / \sum_{i=k}^{\infty} P_{ki} = P_{kj}/WP_k \tag{1}$$

The expected paid premium received in month j from premiums written during month k is $WP_k\ p(j-k)$. If we consider the written premiums in two consecutive months k, k+1, the associated expected paid premium received during month j is:

$$WP_k p(j-k) + WP_{k+1} p[j-(k+1)]$$

In general, the expected paid premium received during month j, EP_j, as a result of writing insurance in all previous periods is

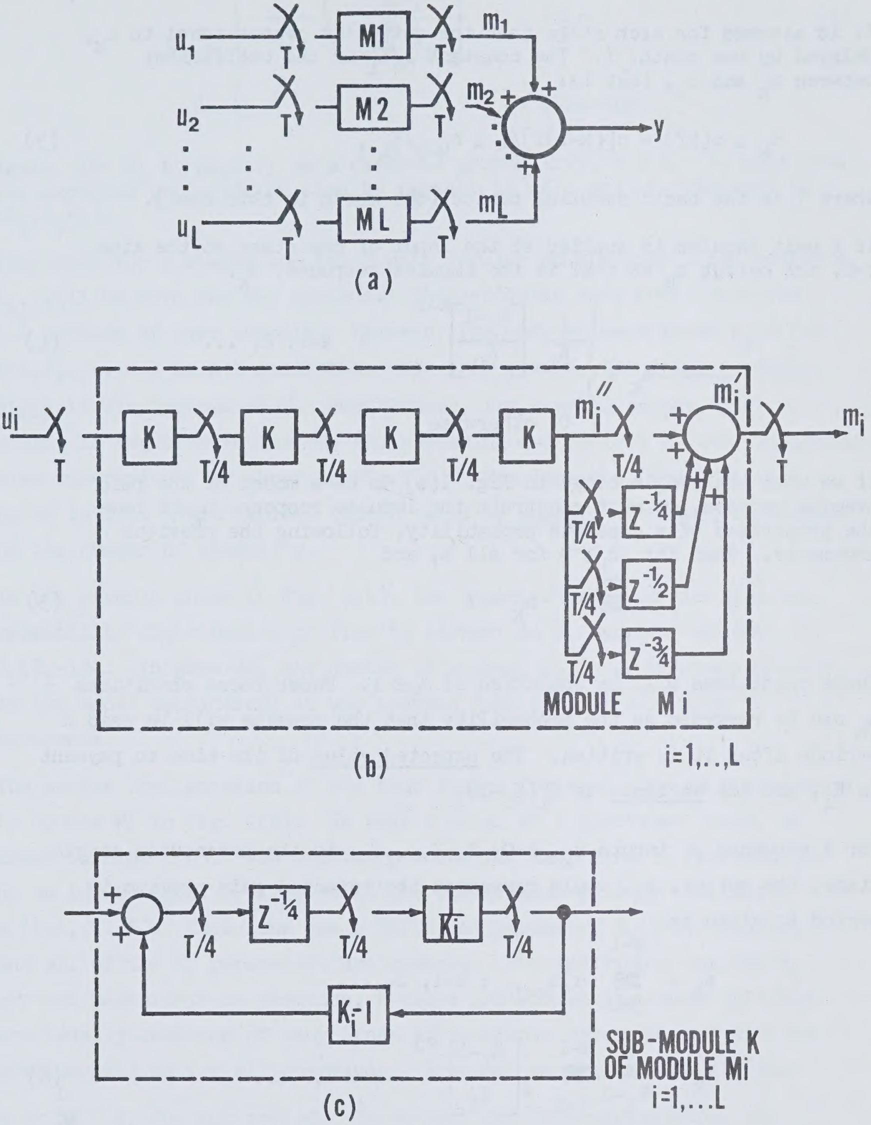
$$EP_j = \sum_{k=0}^{j-1} WP_k p(j-k) \tag{2}$$

The expected paid premium time series, resulting from a written premium time series, can be regarded as the convolution of the written premium series with the probability of the time to payment series. Since the expected paid premium series can be expressed as a convolution of a discrete probability with the written premium series, we can say that a system whose output is a convolution of an impulse response with the applied input can be considered as a potential model for the paid premium process. In particular, a linear sampled-data system has these properties.

If the discrete probability of the time to payment were known, the modeling of this process would be reduced to synthesizing a system from a given impulse response. In practice this probability is usually not specified. Moreover, there usually are no measurable data which would permit its direct evaluation.

In modeling insurance premium payments, we assumed a linear process consisting of a number of stages connected in series. A block diagram of a single stage is shown in Fig. 1(c). (Assume for the moment that instead of $z^{-\frac{1}{4}}$ and T/4, we have a whole period delay z^{-1}, and sampling period T, respectively.) We will designate the stage input as u_k and its output as z_k with k representing the discrete time index. The first stage represents all insured, paying their premiums to insurance agents. Its input is the written premium and the output is premiums paid to agents. The variable $c_k = u_k + (K_i-1)z_k$ represents all the premiums that have been written, but not yet paid to the agent. Index i represents a particular line of insurance.

The second stage represents insurance agents who collect premiums from the insureds and in turn, send them to the insurance company. Stage three represents the receiving, recording and accounting processes within the company. In fact, there may be additional stages representing the mailing and delivery of premiums between each of the three preceeding



u_1
u_2
u_L
T
M1
M2
ML
m_1
m_2
m_L
+
y
(a)
ui
K
T/4
m_i''
m_i'
m_i
$Z^{-1/4}$
$Z^{-1/2}$
$Z^{-3/4}$
MODULE Mi
i=1,..,L
(b)
K_i
K_i-1
SUB-MODULE K
OF MODULE Mi
i=1,...L
(c)

FIG. 1 BLOCK DIAGRAM OF THE SYSTEM

stages. In general, the total number of stages is a variable. In the model adopted in this paper, four stages were chosen for each line of insurance, as shown in Fig. 1(b).

It is assumed for each stage that the output is proportional to c_k, delayed by one month, T. The constant $1/K_i$ is the coefficient between z_k and c_k, that is:

$$z_k \doteq z(kT) = c[(k-1)T]/K_i \doteq c_{k-1}/K_i \ , \tag{3}$$

where T is the basic sampling period (one month in this case).

If a unit impulse is applied at the input of one stage at the time $t=0$, the output z_k at $t=kT$ is the impulse response, h_k.

$$h_k = \begin{cases} \dfrac{1}{K_i}\left[\dfrac{K_i-1}{K_i}\right]^{k-1} & ; \quad k=1,\ 2,\ \ldots \\ 0; \text{ otherwise} & \end{cases} \tag{4}$$

If we want the single stage in Fig. 1(a) to be a model of the paid premium process, we must constrain the impulse response h_k to have the properties of a discrete probability, following the previous arguments. That is: $h_k \geq 0$ for all k, and

$$\sum_{k=0}^{\infty} h_k = 1 \tag{5}$$

These conditions will be satisfied if $K_i \geq 1$. Under these conditions h_k can be regarded as the probability that the premium will be paid k periods after it is written. The expected value of the time to payment is K_i, and the variance is $K_i(K_i-1)$.

For a sequence of inputs u_k, $k=0,\ 1,\ 2,\ \ldots.$, to the preceeding single stage, the output, z_k, would represent the expected paid premium in period k, given by

$$z_k = \sum_{j=0}^{k-1} u_j h_{k-j} \quad ; \ k=1,\ 2,\ \ldots$$

$$z_k = \frac{1}{K_i-1}\sum_{j=0}^{k-1} u_j \left[\frac{K_i-1}{K_i}\right]^{k-j} \quad ; \ k=1,2,\ldots \tag{6}$$

If we had n of the preceeding stages in series, each stage having the same K_i, the corresponding impulse response would be

$$h_k = \begin{cases} \dfrac{1}{K_i^n} \begin{pmatrix} k-1 \\ n-1 \end{pmatrix} \left[\dfrac{K_i - 1}{K_i} \right]^{k-n} & ; k \geq n \\ 0 & ; \text{otherwise} \end{cases} \tag{7}$$

Again, for h_k to qualify as a discrete probability, $K_i \geq 1$. In this case the expected value of the time to payment is nK_i, and the variance is $nK_i(K_i-1)$.

Note that for n stages and for a unit impulse at the input, the response, h_k, will be zero for n-1 periods. Theoretically this model produces n-1 periods of zero outputs. However, for $n>2$, an associated $h_k=0$ for $k=1,2,\ldots,n-1$ is not realistic. For this reason, the delay of each stage is T/n instead of T. Furthermore, the overall output after n stages is required to include the preceding n-1 outputs of the fractional time sub-periods, as shown in Fig.1(b). For instance, when n=4, the delay is T/4. This implies that h_k will be nonzero for $k \geq 1$, regardless of the number of stages, n.

In the example shown in Fig. 1(b), the expected value of the discrete probability distribution of time to payment is K_i and the variance is $K_i(K_i-1)4$. In general, the number of stages, n, is an unknown. However, in the model considered, it was assumed that n=4 for all lines of insurance.

The series configuration of the four stages for each line is represented by module Mi in Fig. 1(b). We have a total of L insurance lines, as shown in Fig. 1(a). The written premium for each line is represented by an input signal $u_i (i=1,\ldots,L)$ and the premium paid by the output $m_i (i=1,\ldots,L)$. Each line has a different parameter $K_i (i=1, \ldots, L)$, and all of the K_i parameters are unknown. The individual outputs m_i are not measurable in practice, because individual insurance policies are usually composed of many lines of insurance coverage, yet the one premium bill is for all coverages. However, we can measure the sum $y = \sum_{i=1}^{L} m_i$ for any period. To account for this aggregation, the premium payment process model is hypothesized as figure 1(a). Based on a sequence of measurements $y(j)$, $(j=1, 2, \ldots, N_p)$, the K_i parameter values are estimated. This will be described in the next section.

III ESTIMATION OF PARAMETERS

As we can see in Fig. 1(a), there are L modules Mi, whose inputs are $u_i(j)$ and whose outputs are $m_i(j)$, ($i=1, \ldots, L$; $j=1, \ldots, N_p$). Each module, Mi, depends on a parameter K_i. All of the K_i parameters are unknown apriori and are to be estimated. It should be noted that all Mi modules have the structure shown in Fig. 1(b), (c). They differ only in the values of the K_i parameters. For each module Mi, given the input sequence $u_i(j)$ and assuming a specific value of K_i, we can always calculate the output sequence $m_i(j)$, based on the system configuration given in Fig. 1(b),(c). Appropriate formulas for the computation of the $m_i(j)$ sequence, given the input sequence $u_i(j)$ and the parameter K_i, are shown in Fig. 2. For convenience of notation, the $R_i=1/K_i$ parameter is used. These formulas apply to each one of the Mi modules. It should be noted that these formulas are of the fourth order with respect to R_i(or $1/K_i$). Moreover, since the system of figure 1(b) has double sampling (T and T/4) and a multiple delay network at the output, we have to evaluate four consecutive expressions while going from $m_i(j)$ to the next output, $m_i(j+1)$. This fact presents a definite difficulty in evaluating the K_i parameters. Another difficulty, mentioned in section II, is the fact that in practice, the outputs $m_i(j)$ are not directly measurable. Only their sum, $y(j) = \sum_{i=1}^{L} m_i(j)$ is measurable for each j. Attacking this problem directly, one could consider a set of consecutive measurements of y(j) and then solve a set of complicated nonlinear equations for K_i. This approach does not seem practical. The approach used in the present application is the following.

We have shown that for each month the output m_i, Fig. 1(b), of each line of insurance can be thought of as an estimate of the expected value of the premiums paid for that line and month. Furthermore, since the expected value of a sum of random variables is the sum of the expected values of each random variable, we can think of our discrete sampled data system, Fig. 1(a), and its associated defining equations, as a mathematical model of the expected aggregate paid premiums of all lines of insurance. In fact, estimating the parameters ($K_1 \ldots K_L$) of the model is really a problem in non-linear regression where the dependent variable is the aggregate paid premium (say O), the independent variables are u_i, ($i=1, \ldots L$), and the parameters are K_i,($i=1 \ldots L$.) We find the

$$m''_i(j) = R_i^4 u_i(j-1) + 4(1-R_i)m''_i(j-1/4) - 6(1-R_i)^2 m''_i(j-1/2) + 4(1-R_i)^3 m''_i(j-3/4) - (1-R_i)^4 m''_i(j-1)$$

$$m''_i(j+1/4) = R_i^4 u_i(j-1) + 4(1-R_i)m''_i(j) - 6(1-R_i)^2 m''_i(j-1/4) + 4(1-R_i)^3 m''_i(j-1/2) - (1-R_i)^4 m''_i(j-3/4)$$

$$m''_i(j+1/2) = R_i^4 u_i(j-1) + 4(1-R_i)m''_i(j+1/4) - 6(1-R_i)^2 m''_i(j) + 4(1-R_i)^3 m''_i(j-1/4) - (1-R_i)^4 m''_i(j-1/2)$$

$$m''_i(j+3/4) = R_i^4 u_i(j-1) + 4(1-R_i)m''_i(j+1/2) - 6(1-R_i)^2 m''_i(j+1/4) + 4(1-R_i)^3 m''_i(j) - (1-R_i)^4 m''_i(j-1/4)$$

$$m''_i(j+1) = R_i^4 u_i(j) + 4(1-R_i)m''_i(j+3/4) - 6(1-R_i)^2 m''_i(j+1/2) + 4(1-R_i)^3 m''_i(j+1/4) - (1-R_i)^4 m''_i(j)$$

$$R_i = 1/K_i; \; i = 1, \dots, L$$

$$m'_i(j) = m''_i(j) + m''_i(j-\tfrac{1}{4}) + m''_i(j-\tfrac{1}{2}) + m''_i(j-3/4)$$

$$m_i(j) = \begin{cases} m'_i(j) & ; \; j \text{ is integer} \\ 0 & ; \; \text{otherwise} \end{cases}$$

Figure 2: The Transfer function of Module Mi

"best" parameters by minimizing the following squared error criterion, J, with respect to the parameters.

$$\text{Minimize} \quad J = \sum_{j=0}^{N_p} \left[\sum_{i=1}^{L} m_i(j) - O(j) \right]^2 \tag{8}$$

Where $O(j)$ = actual output measured at $t=jT$

N_p = total number of periods considered

$m_i(j)$ = output of module Mi calculated on the basis of a known sequence of inputs u_i and an assumed parameter K_i, at $t=jT$. $(i=1,\ldots,L)$

This unconstrained minimization problem was solved using the Nelder and Mead method [2, ch. 4], which does not require explicit evaluation of derivatives. Computational examples are given in the next section.

IV. EXAMPLES

In order to test the proposed methodology, a number of experiments were conducted with simulated data. Aggregate premium payment data was generated by the model using known values of K_i. The experimentation consisted of applying the technique to obtain the estimates of K_i. Estimated values of K_i were then compared against the known K_i. The experimentation was conducted for various sizes of the problem, and both the accuracy of the estimates and the computation time are presented below.

The specific steps undertaken to evaluate the methodology proposed were as follows:

(1) A number of lines L, the number of periods N_p, and a set of parameters $K_1, K_2, \ldots, K_L$ were chosen. The K_i parameters are components of an L-dimensional vector $\underline{K}$.

(2) All inputs were chosen as step functions, starting from $t=0$, and their values $u_i (i=1,\ldots, L)$ were specified.

(3) Based on the data and the mathematical model, the outputs $m_i(j)$, $(i=1,\ldots, L;\ j=1,\ldots, N_p)$ and their sums

$$y(j) = \sum_{i=1}^{L} m_i(j) \quad ; \quad j=1,\ldots,N_p$$

were calculated. This set of $y(j)$ was used as $O(j)$ in the formula in Eq.(8) in the Nelder and Mead minimization procedure. Let $\underline{O} = [O(1), O(2),\ldots,O(N_p)]^T$.

(4) Assuming an initial set of parameters $K_{10}, K_{20}, \ldots, K_{L0}$, the Nelder and Mead program was run to estimate the $K_1, K_2, \ldots, K_L$ parameters which were compared to those assumed in paragraph (1).

All computations were performed on the UNIVAC 494 computer. The Nelder and Mead code was taken from a listing in Reference [2]. (The program worked immediately, according to the listing given in [2]. No debugging was necessary.) The solved examples are listed in the following:

Example 1: $L=2$; $N_p = 3$

$\underline{K} = [2 \quad 4]^T$

$u_1 = u_2 = 100$; $\underline{O} = [6.64 \quad 75 \quad 128]^T$

Results:

Parameter	Initial Guess	Solution
K_1	1.5	2.00005
K_2	3.5	4.00006

Convergence was attained in 10.1 seconds. The STEP parameter, utilized by the Nelder and Mead program [2] was 0.5 in this case. The same problem was run using the Fiacco and McCormick SUMT algorith option which does not use derivatives [3]. This required over 4 minutes for convergence.

Example 2: $L = 3$; $N_p = 4$

$\underline{K} = [1 \quad 2 \quad 4]^T$

$u_1 = u_2 = u_3 = 100$; $\underline{O} = [107 \quad 175 \quad 228 \quad 258]^T$

Results:

Parameter	Initial Guess	Solution
K_1	1.5	0.999995
K_2	2.5	2.000100
K_3	3.5	3.999930

Computing time = 19.4 seconds ; STEP = 0.5

Example 3: $L=6$; $N=7$

$\underline{K} = [1 \quad 1.5 \quad 2 \quad 2.5 \quad 3 \quad 4]^T$

$u_i = 100$; $i=1,\ldots,6$;

$\underline{O} = [130 \quad 333 \quad 466 \quad 535 \quad 570 \quad 586 \quad 594]^T$

Results:

Parameter	Initial Guess	Solution
K_1	1.2	1.00028
K_2	1.7	1.49718
K_3	2.2	2.00418
K_4	2.7	2.49706
K_5	3.2	3.00050
K_6	4.2	4.00015

Computing time = 3 min, 31.1 sec ; STEP = 0.5

Example 4: $L=21$ (a "real life" number); $N_p = 22$; $\underline{K} = [1 \quad 1.2 \quad 1.4 \quad 1.5 \quad 1.7 \quad 1.8 \quad 2 \quad 2.2 \quad 2.4 \quad 2.5 \quad 2.7 \quad 2.8 \quad 3 \quad 3.3 \quad 3.5 \quad 3.8 \quad 4 \quad 4.2 \quad 4.5 \quad 4.7 \quad 5]^T$

The program required 30 minutes to converge to within ±0.01 limits of the parameters values.

V CONCLUSION

The basic practices of discrete-time systems have been demonstrated to be applicable in the analysis of a certain class of financial problems of insurance. An algorithm for estimating model parameters was developed and tested on a number of examples. The potential reduction in the cash flow forecast errors amply justifies the timely application of the proposed technique to update the process parameters in this insurance application.

REFERENCES

1. B.C. Kuo, "Discrete-Data Control Systems", Prentice-Hall, Englewood Cliffs, New Jersey 1970.
2. D.M. Himmelblau, "Applied Nonlinear Programming", McGraw Hill, New York 1972.
3. A.V.Fiacco, G.P.McCormick, "Nonlinear Programming: SUMT", Wiley, New York 1968.

MATHEMATICAL MODELING OF AN AQUATIC ECOSYSTEM I

NARESH N. GUPTA
Wright State University
Dayton, Ohio

JAMES HOUDESHELL
Inland Division
General Motors Corporation
Dayton, Ohio

ABSTRACT

This paper presents a mathematical model of an aquatic ecosystem. The model differs from previous models in that it incorporates characteristics of specific phyla of phytoplankton and zooplankton, inter and intra phyla interaction and limiting nutrient processing by decomposers.

INTRODUCTION

The science of ecology is "the study of systems at a level in which individuals or whole organisms may be considered elements of interaction, either among themselves, or with a loosely organized environmental matrix" [1]. Systems at this level are called ecosystems, and one of the most fundamental and widely studied is an aquatic ecosystem. The majority of aquatic organisms in terms of biomass can be classified as planktonic in nature. Directly or indirectly, all open-water life is dependent upon photosynthetic plankton in the upper level of water illuminated by sunlight. Photosynthesis is the main mechanism for converting energy entering a lake into living cells which then support other life forms and affect the properties of the lake [2]. Lakes of high productivity exhibit large fluctuations in the populations of planktonic organisms. These fluctuations are brought about by functional interdependencies between the biota and their environment. Because of the complexity of these interdependencies, the increased use of mathematical modeling has been necessary. Modeling has been shown to be important in determining limiting factors in executing sound resource management policies [3], and in discovering relationships about the dynamic aspects of ecosystems, which are not apparent from steady-state solutions [4].

To understand aquatic ecosystems, two approaches have been used: the "black box", macroscopic view of a water system; or a microscopic view of each of the components. The "black box" can be visualized as a volume of lake water (figure 1) with predictable inputs of temperatures, photoperiod, light intensity, and specific concentrations of materials included with the flow. These materials are acted upon by the biological material and the outputs are in the form of sediments, and different concentration of materials in the departing flow. There are also unexpected inputs due to man-made or natural accidents, such as herbicides or sewage. By following slug inputs, the characteristics of the system can be determined. This is the approach taken by most water control engineers.

Much less work has been done on the microscopic approach in determining the dynamics contained within the "black box". Primarily because the real world turns out to be nonlinear in character, and stochastic in predictability. In formulating a complete model of the microscopic approach, a review of previous work is necessary, beginning with the classic work by Lotka and Volterra [5, 6]. They theorized that population growth rates are exponential in nature:

$$\frac{dN}{dt} = rN \qquad N_t = N_o e^{rt} \tag{1}$$

expanding equation one by including the environmental resistance [1-(K-N)/K] one derives a logistic growth curve [7].

$$\frac{dN}{dt} = N - \frac{K-N}{K} \tag{2}$$

where K is the carrying capacity of the environment. From this basic beginning the models become more complex by the inclusion of feedback and the use of simultaneous equations. B. C. Patten [8] has written an exhaustive review of mathematical models of plankton productivity. This review indicated that all of the models constructed before 1968 may be classified as one of two types. The first type of model calculated biomass as a function of time only. A typical example is that of Davidson and Clymer [9] where an analog simulation was made using the equations:

Temperature	$T = K_9 - K_{10} \text{ vs } \frac{\pi t}{6}$
Incident Radiation	$I = K_{11} - K_{12} \text{ vs } \frac{\pi t}{6}$
Nutrient Conc.	$\overset{\circ}{N} = K_7 - K_8 P \quad N \geq 0$
Phytoplankton	$\overset{\circ}{P} = (K_1 N^{cim} IT - K_2 Z - K_3 - K\pi)P$
Zooplankton	$\overset{\circ}{Z} = (K_5 P - K_6) Z$

The second type of model, which considers plankton population as a function of depth only, was developed by Murphy [10]. This equation was

$$\frac{dP}{dZ} = I_e e^{-KZ} - 4.3$$

where I_e is the photosynthetically effective energy penetrating the water surface.

Until recently the problem of modeling plankton populations as functions of both time and depth had remained unsolved. A preliminary simulation was developed by Bungay and Abernathy [11] and more recently by Mitchell [12]. This leads up to the authors' model which departs from previous studies.

DESCRIPTION OF THE MODEL

The authors' model falls under Patten's group one, by considering productivity as a function of time. However, this model differs from previous models by the inclusion of specific phyla of phytoplankton and zooplankton by considering interphyla inhibition, by considering more than one possible limiting nutrient, and by including time lags in nutrient processing by decomposers. The general group of phytoplankton has been broken down into three representative phyla:

(1) Myxophyceae (Blue-green algae)
(2) Chlorophyceae (Green algae)
(3) Chrysophyta Class Bacillarieae (Diatoms)

Each phyla has its own characteristic differences. For example, the diatoms require silicates to form frustules (shells), thus silicate is a possible growth limiting factor [13]. Just as the phytoplankton have been broken down into three representative phyla, so have the zooplankton been divided into two major groups:

(1) Phylum Rotifera
(2) Phylum Arthropoda Class Crustacea

Both of these groups have major differences in growth rates, life expectancy and feeding habits. The first group Rotifera is characterized as a small herbivore approximately four hundred microns in length [14]. Class Crustacea can be characterized by Dapnia which is an omnivore approximately one millimeter in length [14].

The inclusion of inter and intra phyla inhibition is accomplished by the inclusion of a term $(K_i - S_i)/K_i$ as a limiting factor affecting the growth potential of the species. This brings out a sigmoid shaped curve. Differential equations can be written to govern the dynamics of the community. These solutions can be characterized as exponentials similar to equation (2). In the model this was simulated by breaking the continuous time equation into algebraic equations with time increments of one week in nature.

In discussing the mathematical formulation of the model it will be necessary to break down the model into four characteristic groups (figure 2);

environmental, producer, consumer and decomposer. The environmental section contains factors pertaining to climatic conditions and nutrient levels. These equations are:

(1) Temperature $\quad ETE = \{1 + 20 \text{ Exp } [-0.5(t-30)/9]^2\}/21$

(2) Flow/Vol $\quad EFL = 1.3 + 0.5 \text{ Exp } [-1/2(t-40)/6]^2 +$
$\quad\quad 0.18 \text{ Exp } [-1/2\,(t-14)/4.5]^2$

(3) Photoperiod $\quad EPH = \{12.2 + 3.5 \sin [(38-t)/26]\}/15.7$

where t = time in weeks.

Also involved are three equations governing nutrient levels. These equations describing nitrate, phosphate and silicate levels have the general form:

$$N_i(t + 1) = N_i(t) + (C_i \text{ in} - C_i \text{ out}) \, EFL + D_i \text{ Transformed } (t) -$$
$$\sum_{j=1}^{3} E_{ij}[P_{ij}(t) - P_{ij}(t-1)] - L_i \cdot N_i(t); \quad N_i(t + 1) \geq 0$$

and the producer section can be generalized in the form

$$P_i(t + 1) = \text{Exp } [r_i(b_i - m_i)]P_i(t)\{1 - \sum_{j=1}^{2} [C_{ij}(t) - C_{ij}(t-1)] - L_i\}$$

where b is the reproductive rate and m is the death rate. B is a function of temperature, photoperiod, nutrient levels, inhibition by inter and intra phyla interaction and light intensity; m is a function of temperature; the term $\sum_{j=1}^{2} [C_{ij}(t) - C_{ij}(t-1)]$ is phytoplankton material required by consumers for growth.

Consumers can be generalized in the form:

$$C_i(t+1) = \text{Exp } [r_i(b_i-m_i)]C_i(t)\{1 - \sum_{j=1}^{2} [C_{ij}(t) - C_{ij}(t-1)] - L_i\}$$

where b_i is a function of food levels, temperature and nutritional value of nutriment and m_i is a function of temperature. The term $\sum_{j=1}^{2} [C_{ij}(t) - C_{ij}(t-1)]$ may or may not be present depending on whether or not the consumer is in turn consumed. The fourth and final group is the decomposer which can be thought of as the effects of bacteria upon transforming dead material into usable nutriment. The general form of this group is

$$D_i(t+1) = C_i \text{ in} - C_i \text{ out}) \, EPC + \sum_{j=1}^{2} [A_j P_j(t) + b_j C_j(t)] -$$
$$d_i \, ETE \cdot [D_i(t) + D_i(t-1)]; \quad D_j(t+1) \geq 0$$

CONCLUDING REMARKS

Although the present study is not completed, several points have been shown. As one moves up the trophic levels, the senstivity of the model to fluctuations in biomass at the top layer becomes greater. This confirms Garfinkel's [15] supposition that quadratic density-dependent terms tend to control fluctuations. Another preliminary result is that diatom growth is limited by silicate concentrations in this model. In addition a doubling of the growth rate of plankton has little effect on the time in which peaks occur or in the magnitude of the peaks [16]; it does however make the biomass fluctuate with greater amplitude around the carrying capacity.

After the model is refined a test case will be tried. Presently, the western basin of Lake Erie is the prime choice. Other ideas which could be incorporated into the model are the type of distributions of phytoplankton and zooplankton in the water. In our model a uniform distribution is assumed; however, phytoplankton are known to grow in clusters. This has an effect on the type of zooplankton grazing [17]. Other significant factors are interactions between the benthic and plankton communities, the fluctuations from weather, the effect of wind on turbidity and thus light intensity. The inclusion of micronutrients such as cobalt or molybdenum or the effects of pH, hardness and distribution of metal salts concentrations could be considered.

REFERENCES

[]] Margelef, R., Perspectives in Ecological Theory, Univ. of Chicato Press, 1968.

[2] Hutchinson, G. E., A Treatise on Limnology, Introduction to Lake Biology and the Limnoplankton, New York: Wiley, 1957.

[3] Watt, K. E. F., Ecology and Resource Management, New York: Wiley, 1968.

[4] Brennan, R. D., Dewit, C. T., Williams, W. A., Quattrin, E. V., "The Utility of a Digital Simulation Language for Ecological Modeling," Oecologia (Beri.), 4, pp 113-132, 1969.

[5] Lotka, A. J., "The Structure of a Growing Population," Human Biology, 3, pp 459-493, 1931.

[6] Volterra, V., "Variations and Fluctuation of the Number of Individuals in Animal Species Living Together," in R. N. Chapman (Ed.) Animal Ecology, New York: McGraw Hill, 1931.

[7] Boughey, A. S., Ecology of Populations, New York: MacMillan Co., 1968.

[8] Patten, B. C., "Mathematical Models of Plankton Productivity," Int. Revue Ges. Hydrobiol., 53, pp 357-408, 1968.

[9] Davidson, R. S., Clymer, A. B., "The Desirability and Applicability of Simulating Ecosystems," Ann. N. Y. Acad. Sci., 128, pp 790-794, 1966.

[10] Murphy, G. I., "Effect of Mixing Depth on the Productivity of Freshwater Impoundments," Trans. Amer. Fish Soc., 91, pp 69-76, 1962.

[11] Bungay, H. R., Abernathy, A. R., Reservoir Water Quality Prediction Based on Limnological Parameters. Clonson Water Quality Res. Research Inst., 1969.

[12] Mitchell, L. W., Digital Simulation of a Limnoplankton Ecosystem, M. S. Thesis, Rose-Hulman Inst. of Tech., 1970.

[13] Lewin, J. C., "Silicon Metabolism in Diatoms, IV Growth and Frustule Formation," in Navicula Pelliculosa Can. J. Microbiol., 3, pp 427-433, 1957.

[14] Pennak, R. W., Fresh-Water Invertebrates of the United States, New York: Ronald, 1953.

[15] Garfinkel, D., "A Simulation Study of the Effect on Simple Ecological Systems of Making Rate of Increase of Population Density-Dependent," Simulation, 6, pp 275-281, 1967.

[16] Parker, R. A., "Simulation of an Aquatic Ecosystem," Biometrics, pp 803-821, 1968.

[17] Ryther, J. H., "Inhibitory Effects of Phytoplankton upon the feeding of Daphniz Magna with reference to Growth, Reproduction, and Survival," Ecology, 35, pp 522-532, 1954.

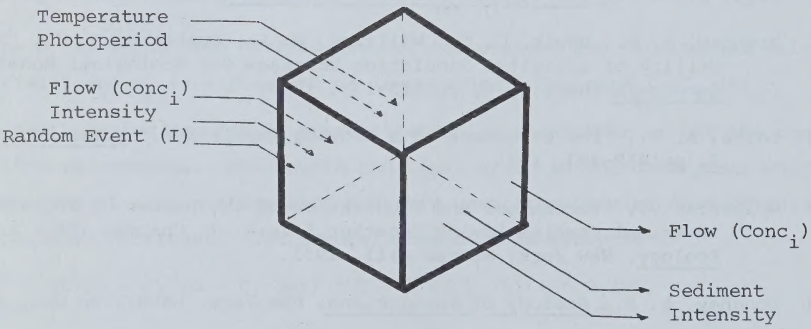


Figure 1

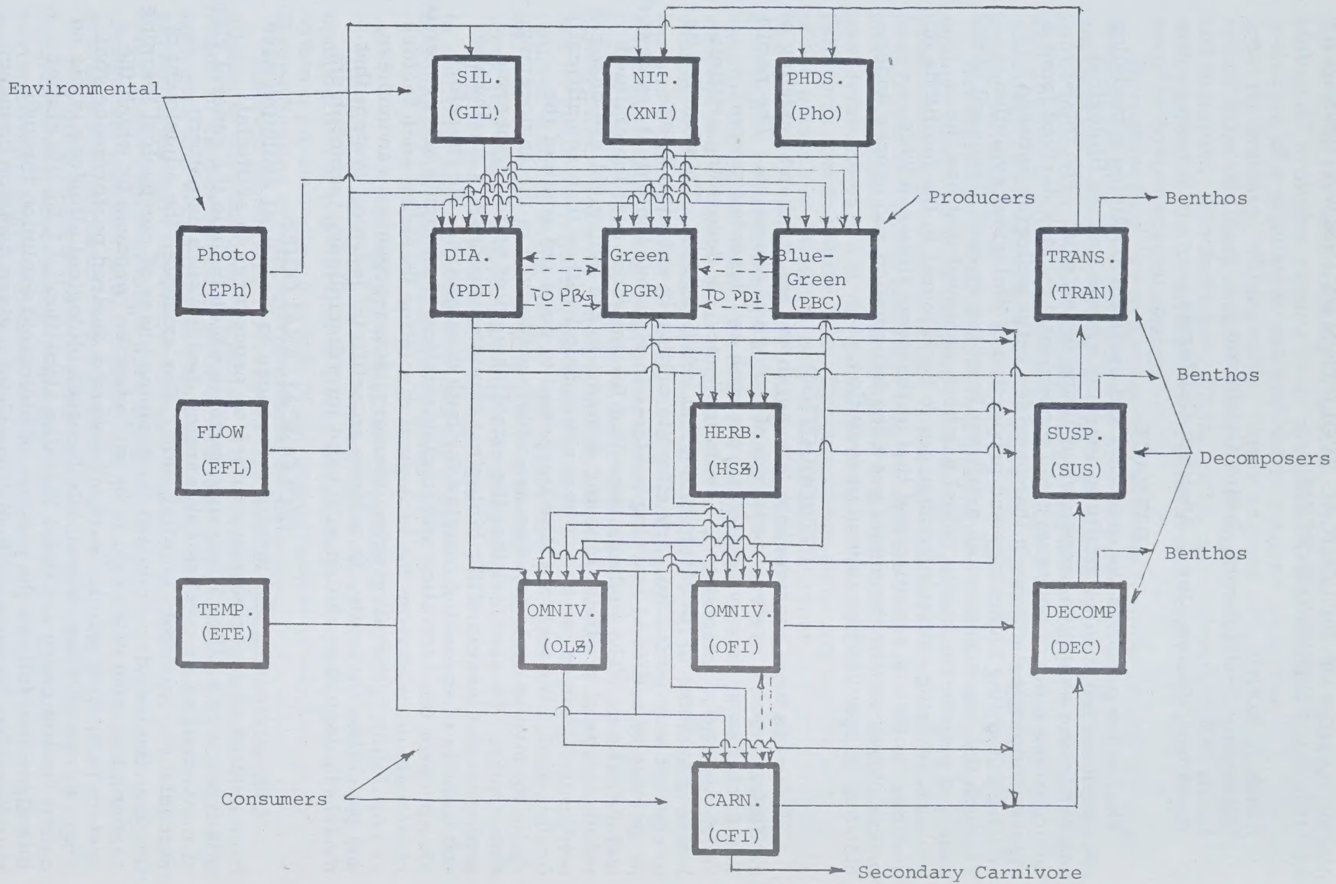

Environmental
SIL.
(GIL)
NIT.
(XNI)
PHDS.
(Pho)
Producers
Photo
(EPh)
DIA.
(PDI)
TO PBG
Green
(PGR)
TO PDI
Blue-
Green
(PBC)
TRANS.
(TRAN)
Benthos
FLOW
(EFL)
HERB.
(HSZ)
SUSP.
(SUS)
Benthos
Decomposers
TEMP.
(ETE)
OMNIV.
(OLZ)
OMNIV.
(OFI)
DECOMP
(DEC)
Benthos
Consumers
CARN.
(CFI)
Secondary Carnivore

Figure 2

A SIMULATION OF BIOLOGICAL EVOLUTION BY INNOVATIONS IN AN ARTIFICIAL LEARNING SYSTEM

SARA J. WAN
University of California, Irvine California
L. A. WAN
President, Sycom, Inc., Anaheim California

ABSTRACT

This paper presents the direction and basis for obtaining a modeling isomorphism between an artificial learning system and the biological adaptation and evolution processes of living organisms. The model incorporates a set of necessary and fundamental factors derived from a rigorous biological approach that provides valid analogies between artificial learning technology and properties of biological evolution. Through the innovations to an artificial learning system discussed in the paper, a macro-stochastic model has been achieved that has the property of being sufficiently abstract to be relevant to the multitude of peculiar factors of evolution and that at the same time has wide isomorphism so that the model provides information and proper understanding of specific phenomena in evolution.

INTRODUCTION

There has been considerable interest in the analysis and modeling of artificial learning systems in terms of biological analogies. The result is a profusion of loosely applied terms such as "learning curves", "goal seeking", "adaptation", "evolution", etc., in describing artificial learning systems to implicate biological relationships. This has led us to conduct an interdisciplinary effort to critically examine these properties of artificial learning systems in rigorous biological framework and definitions. The findings show that for the most part the claimed relationships of artificial learning systems to the biological are cases of poetic metaphorism or are at best coincidental rather than scientifically fundamental. We feel that it is important to point out some of the saliently invalid analogies that are widely publicized in the engineering community. We also feel that the establishment of valid engineering analogies and models of the biological evolution process is worthwhile and capable of expanding insights for both disciplines. We therefore also present the derivation and implementation of elements and properties rigorously derived from the biological discipline including such factors as population, generation time, memory, intra-organism randomness, and population diversity, to achieve an artificial learning system that models biological evolution with valid interdisciplinary isomorphism.

BIOLOGICAL ANALOGIES

Many misleading and invalid references to biological evolution have been applied to describing properties or procedures in artificial intelligence or artificial learning systems which include the general class of mathematical or mechanical learning devices such as pattern recognizers, threshold learning, etc. For example, the adjustment of the decision boundaries during the training phase of mechanical learning systems has been often said to be an "adaptive" process by which the system is further said to "evolve" toward a desired performance goal. Fogel is one of those responsible for such biological allusions in his so called "evolutionary artificial learning algorithm". [1, 2] He claims this algorithm follows the process of biological evolution through mutation when he tests perturbations about a given decision boundary with training inputs and selects the one with the best performance to perturb for the next training sequence. It is most naive to assume that biological evolution results from mutqtions from only the best adapted members of a population and that better adapted offspring will only come from its mutants. Biologically, better adapted offspring may very well result from the least adapted member through genetic recombination and mutation. Furthermore, biologically, most mutations are deleterious with respect to the immediate environment and may be fit only in the event of environmental change.

In engineering, we are concerned with the training of a given single mechanical device. Such a device may be perfectly adaptive in the engineering sense but it has no correspondence to evolutionary adaptation in the biological realm. Biologically, adaptation characterizes the distribution of members of the population as a result of selective advantages and disadvantages. Adaptation involves a population not a single individual.

Bremermann [3, 4] has further propagated this fiction in biological analogy where he also randomly perturbed a successful trial of a linear programming process to obtain "mutated" trials for selection of the next fittest individual and calls this an asexual process. He also at times used the arithmetic average of two best fit trials to obtain the next trial and called it the sexual process. The obvious biological fallacy is that in nature, the offspring is not some arithmetic average of the parents and that not only a single pair of most "fit" are permitted to reproduce. Furthermore, since there is no qualitative difference genetically between sexual and asexual reproduction, it is absurd to ascribe biological characteristics of sexuality to a mechanical system. An even more fundamental error is the assumption that evolution, in dealing with large numbers of genetic possibilities, is confronted with the same optimization problem as engineering systems. It is true that a mechanical learning system must perform some useful function and it is necessary for it to achieve some optimum state within given constraints. However, biological evolution is purely accidental. There are no goals or criteria functions for either defining or deriving some optimal state of living things. Survivors are not optimal solutions derived by some grand scheme, nor are extinct organisms the failures, since there are no distinguishing values placed on fitness, life, or death.

The frequent and common misapplications of biological terminologies to artificial learning systems call for a rigorous and definitive description of evolution in a biological framework.

THE FRAMEWORK OF BIOLOGICAL EVOLUTION

Biological evolution is the dynamic process of change in the genetic composition of populations of living organisms. Evolution occurs within a population as a result of the counteraction of inherent population diversification versus environmental suppression, i. e. natural selection.

Inherent population diversity is the variety of intra-organism gene alleles of chromosomal variants. This diversity of genotypes is the result of cross-breeding, gene recombinations, and mutations (including both gene and chromosomal changes). Mutation is the stochastic perturbation upon the conservative process of heredity, which otherwise would make reproduction a pure duplication system. Mutation introduces change and together with the multitude of possible gene combinations provides the enormous diversity within all natural populations. [5]

Evolution is the result of redistribution of the composition and frequencies of occurrence of the genotypes within a population by the environmental suppression of the diversifying genetic variety. In the great diversity of life forms, some fortuitously possess the phenotypes of traits or properties to escape the suppression, that is to evade death or extinction. The present biological entities are merely a lucky minority of the many extinct former inhabitants. Those which escape are said to be "fit" to the environmental conditions at the time. Those which are suppressed are said to be "unfit". Of course, fit and unfit are not discrete quantum conditions. Environmental suppression affects different organisms in varying degrees. The effect is also time correlated since biological processes possess memory and contributions of past organisms propagate into the future. An unfit organism in breeding before dying will cause certain of its characteristics to remain in the population's gene pool via its offspring. Many generations may pass before the elimination of a recessive mutation from the population.

Environmental suppression imposes a probability of survival or a relative frequency of occurrence on each genotype within a population. In fact, a population can best be described as a statistical distribution of genotypes within a given organism category. The continuous reshaping of this distribution resulting from the opposing action between intra-organism random diversity and suppression of the variational organism forms by a randomly time varying environment is biological evolution. The overtly observable effect is a time shifting of the norm of the population distribution towards adaptation to the present environment; that is predominance of the genotypes of the current generation's most fit phenotypes. However, the existence of the relatively unadapted individuals distributed at various distances from the norm represent the potential for survival and redistribution in the event the environment changes. Thus, a population in which all individuals are not maximally adapted to the immediate environment due to the intra-organism random diversity, possesses individuals who may be more fit it a changing environment and thus allows for redistribution of the population composition in the face of changing environmental conditions.

Evolution is the outcome of the interaction of generations of populations of living organisms with the multitude of environmental conditions. The outcome is random and completely undirected, since it results from the stochastic processes of intra-organims randomness and random environmental conditions. The outcome of the interaction of the totality of living things and the totality of environments is the statistical redistribution of populations of individuals or genotypes. In fact the universe of all biological organisms can be characterized as the near infinite multivariate, multi-modal joint probabilities of all properties and traits of living things. Pronounced or well separated modes within the probabilities represent classifiable categories of organisms. [6] Thus, any treatment or modeling of biological evolution must be at the macro-stochastic level.

AN ARTIFICIAL LEARNING SYSTEM SIMULATING BIOLOGICAL EVOLUTION

Analogies to biological evolution in artificial learning systems are only meaningful if the terminologies used rigorously satisfy the biological definitions. Rather than loosely applying biological expressions merely for poetic description, it is possible by a directed effort which maintains scientific rigor in both disciplines to obtain a valid modeling isomorphism between mechanical learning systems and the biological evolution process. By innovations to an artificial learning system so as to incorporate such biological elements as population, generation, intra-organism randomness, memory, and environmental suppression, a model that validly simulates many characteristics of evolution can be derived. This is illustrated by applying the technique to the threshold learning process (TLP) which simply embodies the fundamental characteristics of artificial learning. [7, 8]

In the TLP, we let the random environment be simulated by the training input signal consisting of a binary information source transmitted through a noisy channel. Thus, the statistical nature of the environment is characterized by the channel output probabilities given the binary information source is either a zero or a one. In keeping with the framework of biological evolution discussed above, we are modeling evolution in statistical terms. The channel output statistics, therefore, are used to simulate either a specific environmental condition confronted by a apecific organism or a collective ensemble of environmental conditions confronting a total population.

The TLP interacts with the channel output by a threshold. (See figure 1) Thus, we let the threshold values simulate the range of variations in genotypes in a particular trait of an organism or of different individuals. We must also introduce the biological factor of population. Thus, the ordinary TLP of deterministic threshold settings must be modified by the innovation of a random threshold generator. The threshold, too, becomes a stochastic variable and the threshold probabilities characterize the statistical distribution or relative frequency of occurrence of genotypes within a population. (See figure 2)

The decision made in comparing a threshold value from the random threshold generator against the channel output signal represents the interaction of the environment with a random sample of an individual from the population. In the binary case, a correct decision is considered fit and false rests and false alarms are misfits. The TLP training feedback policy must therefore simulate the process of environmental suppression of the misfits. In the ordinary TLP, the feedback policy adjusts the threshold value. However, in our modified TLP, the threshold is a random variable so the feedback policy can only act upon the threshold statistics and not on the threshold itself. Both for illustrative simplicity, and for the impirical prevalence of normal distributions in natural phenomena, we can let the threshold be Gaussian. In which case the feedback policy must be concerned with adjusting only the mean and standard deviation. Consider first the threshold mean, μ_κ.

Our feedback policy for adjusting the threshold mean [6] in a manner of simulating the biological adaptation of the population norm is as follows:

$$\mu_\kappa(n) = \begin{cases} \mu_\kappa(n-1) + \Delta\mu \sum_{j=1}^{n-1} \alpha^{n-1-j} \sum_{i=1}^{N} w_i e(i,j)\ , & \text{if } Q \leq \beta\sigma_\kappa(n-1) \\ \mu_\kappa(n-1) + \beta\sigma_\kappa(n-1)\ , & \text{if } Q > \beta\sigma_\kappa(n-1) \end{cases}$$

Each iteration, n, corresponds to the nth. generation, while $w_i e(i,j)$ (where e=0 for correct decision and 1 and -1 for false alarm and false rest, respectively) represents the weighting of the performance of the i-th. random threshold value during the jth. training iteration and combine to biologically simulate the degree of fitness of the i-th. of the N organisms sampled from the jth. generation of the population. $\alpha < 1$, is a trailoff time factor that models the residual contributions of individuals from past generation to the current gene pool. Furthermore, the adjustment of the mean is coupled through the parameter, β, to the threshold standard deviation, σ_κ, which acts as a limiting factor to the rate of movement of the mean. This simulates the biological concept that the population mean adapts to the immediate environment by the redistribution of genotypes as a result of the environmental suppression of relative numbers of misfit organisms within the variational spread of the population. This amount of population diversity is measured by, σ, and represents a limiting factor to the degree the population mean can shift toward adaptation. This brings us to the policy for adjusting σ_k itself.

Our feedback policy for adjusting σ, is as follows [6]:

$$\sigma_k(n) = \sigma_k(n-1) + \sigma_m - R$$

where

$$R = \frac{\gamma}{N}\,\sigma_k(n-1)\sum_{i=1}^{N} e^2(i,\, n-1)$$

The factor R, decreasing the standard deviation at a rate, γ, proportional to the number of misfits sampled from the population, simulates environmental suppression of population diversity. This is opposed by the rate of intra-organism randomness, σ_m, which generates the inherent population diversity.

The training procedure for the modified TLP is to test N thresholds sampled from a random threshold generator with a given threshold probability against random outputs from the noisy channel. The mean and standard deviation of the threshold probability are feedback adjusted according to the policies just described for the next iteration of N samples. The training process of this TLP from iteration to iteration simulates the generation by generation evolution of a population modeled by the threshold statistics under the influence of the environment characterized by the channel output statistics.

CONCLUSION

By following the principles in the scientific framework of biological evolution, it is possible to create a modeling isomorphism between artificial learning and biological evolution as a macro-stochastic process. In the example of the TLP, the model is shown in Figure 2 where evolution is treated in statistical terms with the threshold probability modeling the population statistics and the noisy channel output statistics modeling the stochastic environment. Graphs of these probabilities also provide pictorial presentations of the relative adaptation of the population with respect to the environment. See Figure 3.

The channel output conditional probability density functions given the binary source was a zero or a one are shown as curves A and B, respectively. They respresent the statistical characteristics of the environment; that is, the signal to which an individual (threshold) must respond with a zero or one decision. In the case where the channel corrupts the information with additive gaussian noise, the statistics are defined in terms of the noise mean, μ_v, and standard deviation, σ_v, just as the population (threshold statistics) are defined by μ_k and σ_k. Relative adaptation of the norm of a population can be visualized in terms of the location of the population density with respect to the environment densities. The norm of the population is maximally adapted when $\mu_k = \mu_v + 0.5$ (the threshold value in TLP terms of minimum probability of error). The degree of adaptation of the population as a whole can be visualized in terms of the amount of overlapping regions between the density functions and is thus dependent upon the position of μ_k relative to μ_v as well as the sizes of σ_k and σ_v.

Since evolution is the redistribution of organism populations the computer simulation outputs the population distribution from one generation to the next as the population evolves with respect to the environment. A profile of evolution can thus be presented graphically as in Figures 4 and 5 in terms of the population mean and standard deviation versus number of elapsed generations. For populations that evolve toward adaptation, the population norm approaches threshold values about the point of minimum probability of error which for the conditions shown is 0.75. The two curves in Figure 4 exhibit this behavior for populations initially with an unadapted mean of -0.5. This would be equivalent to a population that is facing a changed environment. Their corresponding standard deviations are shown in Figure 5 where we observe an increase in population diversity as the population becomes more adapted and suffers less environmental suppression. Listed also on Figure 4 are several cases where the populations ended in extinction; that is, the population mean never shifted into the adapted region and the population diversity was suppressed to dying diminishment. The reason is that these populations had less diversification rate ($\sigma_m < 0.1$) then the ones that evolved adaptively. The inherent diversification rates, σ_m, were insufficient to overcome the severe environmental suppression.

These examples exhibit the principle of evolution that the degree to which the population can redistribute is constrained by the amount of variety and the rate of diversification within the population. A population with insufficient diversity will not have enough members who can survive the environmental suppression to evolve an adapted population.

The computer simulation behavior has been compared to generally known and established properties of biological evolution such as evolution as a function of initial population variance, evolution as a function of inherent population diversification rate, evolution as a function of initial fitness and properties of adapted populations. These comparisons have verified and validated the modeling isomorphism achieved between the artificial learning system model and process of biological evolution. Furthermore, the model reveals many phenomena and function relationships among the fundamental factors of evolution which suggest areas for additional and specific investigation in the biological realm itself. These include evolution as a function of memory (residual effects) and the interdependence of the various elements of evolution identified in the model.

The page limitations on the Proceedings prevent inclusion of the voluminous results. However, many are presented at the symposium and more can be found in reference [6] and will be in several future publications.

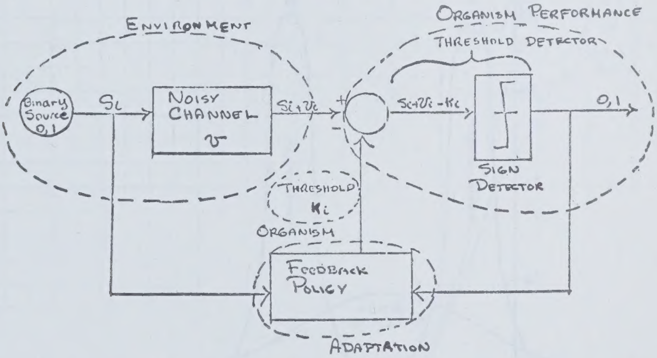
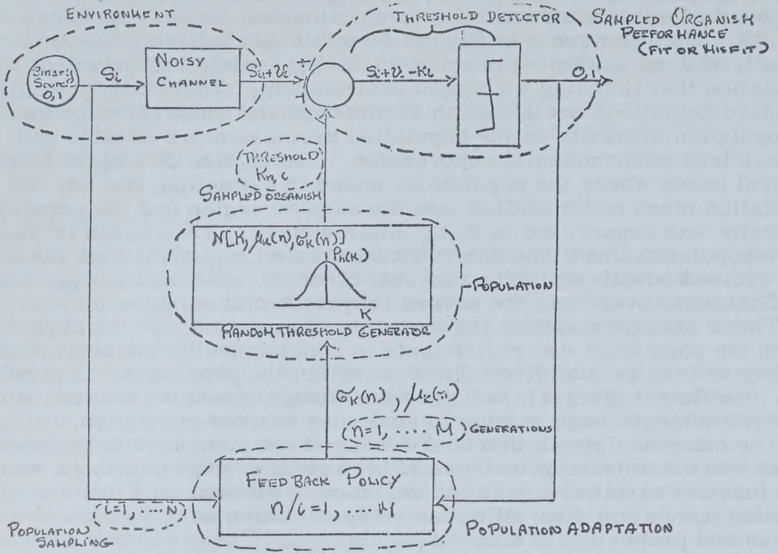


FIGURE 1

TLP CONFIGURATION AND BIOLOGICAL ANALOGIES


Environment
Binary Source 0,1
S_i
Noisy Channel
S_i+v_i
+
−
$S_i+v_i-K_i$
Threshold Detector
Sampled Organism Performance (Fit or Misfit)
0,1
Threshold $K_{n,i}$
Sampled Organism
$N[K, \mu_K(n), \sigma_K(n)]$
$P_n(K)$
K
Population
Random Threshold Generator
$\sigma_K(n), \mu_K(n)$
$(n=1, \ldots M)$ Generations
Feed Back Policy
$n/i=1, \ldots M$
$(i=1, \ldots N)$
Population Sampling
Population Adaptation

Modified TLP with Biological Analogies

FIGURE 2

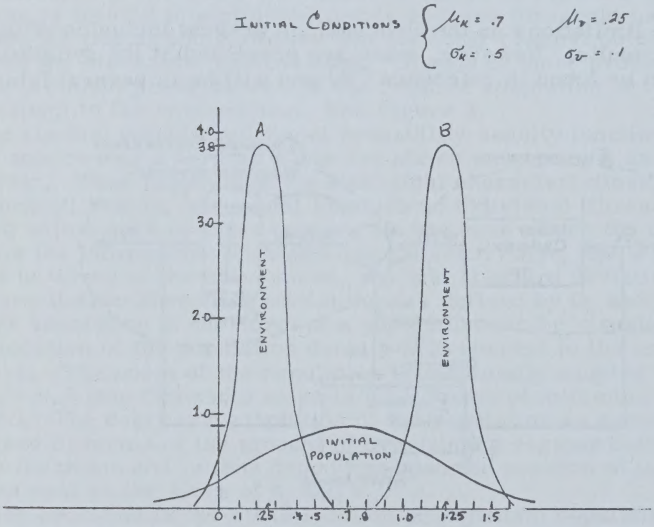


FIGURE 3

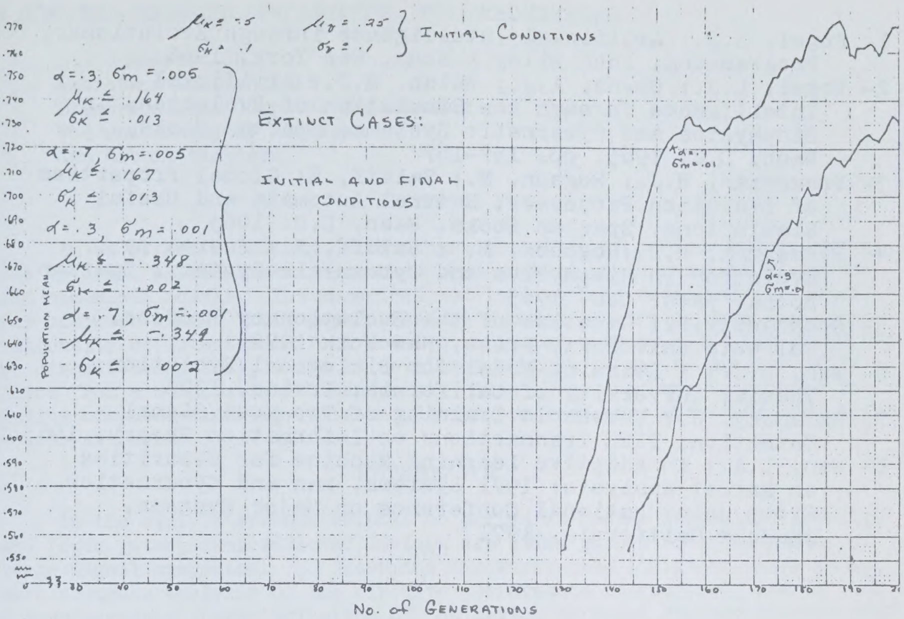


FIGURE 4

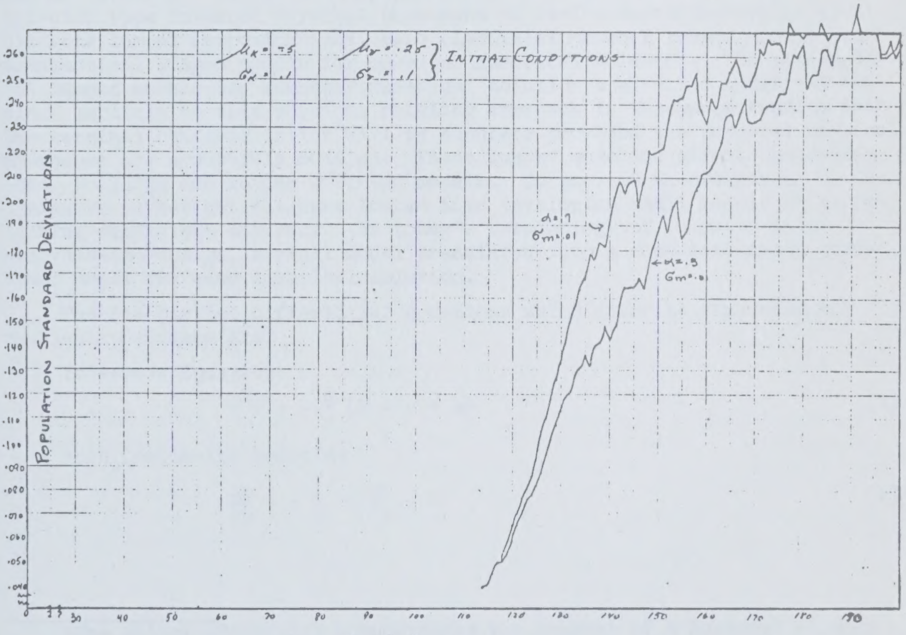


FIGURE 5

REFERENCES

1- Fogel, L.J.; Artificial Intelligence Through Evolutionary Programming, John Wiley & Sons, New York, 1966

2- Fogel, L.J.; Owens, A.J.; Walsh, M.J.; Artificial Intelligence Through the Simulation of Evolution, in Biophysics and Cybernetic Systems, Spartan Books, Wash. D.C. 1965, pp. 157-167

3- Bremerman, H.J.; Rogson, M.; Salaff, S; Global Properties of Evolution Processes; Natural Automata and Useful Simulations, Spartan Books, Wash. D.C. 1965

4- Bremerman, H.J.;Rogoson, M. ; Salaff, S.; Search by Evolution in Biophysics and Cybernetic Systems, Spartan Books, Wash. D.C. 1965

5- Dobshansky,T.; Genetics of the Evolutionary Process, Columbia University Press, New York 1970

6- Wan, S.J.; A Learning Model for Biological Evolution; Thesis, University of California, Irvine, 1972

7- Sklansky, J.; Threshold Training of Two-Mode Signal Detection; IEEE Transactions on Information Theory, 1965

8- Wan, L.A.; An Adaptive Learning Machine for Securities an Market Analysis; 1971 Systems, Man and Cybernetics Group Joint National Conference on Major Systems, Anaheim California, 1971

A COMPUTER-GENERATED SEMICONDUCTOR MODEL*

L. C. McAFEE, JR.
Assistant Professor of
Electrical and Computer Engineering
The University of Michigan
Ann Arbor, Michigan

ABSTRACT

In an effort to obtain better characteristics of semiconductor devices, the internal physical phenomena have been modeled by lumped-network models and numerical models. The numerical models have been made quite efficient for computer usage. This paper presents a model conversion from the computationally efficient numerical model to a lumped-network model (as preferred by circuit designers), which is developed so that it may be equally efficient from a computational standpoint. Also, presented are techniques for making computer-aided-analysis-and design programs more efficient when these models are used.

INTRODUCTION

In the ever determined efforts to arrive at faster switching circuits and logic gates, semiconductor devices are being pushed to their limits of large-signal operation. To determine the worst-case limits of these gates, more accurate analysis of the circuits performance is required, and in turn a more accurate characterization of the device terminal voltage-current characteristics is required. The investigations to arrive at more accurate models of the device terminal characteristics have turned to modeling the distributed type internal physical phenomena of semiconductor materials with discrete lumped approximations, both electrical network lumped models and mathematical lumped models for computer numerical solution. The mathematical lumped models for computer numerical solution are called numerical models. Implicit to this physical modeling approach is the understanding that the terminal characteristics will be accurate provided the internal physical phenomena are accurately modeled. These lumped modeling efforts generally lead to large and rather unwieldy models. To prevent an overburden on the designer, investigations have looked into developing small models while retaining sufficient accuracy. In today's terminology of computer-aided-analysis-and-design, a small model translates into a computationally efficient model for easy numerical solution.

The six partial differential equations which describe semiconductor physical phenomena are:

Poisson's Equation

$$\nabla^2 v = - \frac{q}{\epsilon} (N - n + p) \tag{1}$$

Hole Continuity Equation

$$\frac{\partial p}{\partial t} = - \frac{1}{q} \nabla \cdot \vec{J}_p - R \tag{2}$$

---

*The author gratefully acknowledges the support of a National Science Foundation Research Initiation Grant, No. GK-27886, which is being sponsored through the office of the Division of Engineering, and also support of a Horace H. Rackham School of Graduate Studies Faculty Research Grant.

Hole Current Density

$$\vec{J}_p = - q\mu_p \, p \, (\nabla v + \frac{d\xi_p}{dp} \nabla p) \tag{3}$$

Electron Continuity Equation

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \vec{J}_n - R \tag{4}$$

Electron Current Density

$$\vec{J}_n = - q\mu_n \, n \, (\nabla v - \frac{d\xi_n}{dn} \nabla n) \tag{5}$$

Total Current Density

$$\vec{J} = \vec{J}_p + \vec{J}_n - \epsilon \frac{\partial}{\partial \tau} (\nabla v) \tag{6}$$

The variables used in the above equations, along with variables used throughout this paper are defined in Table I.

Table I

v	voltage	p	hole concentration
n	electron concentration	N	impurity distribution
A	cross-sectional area	R	recombination rate
q	electrostatic charge	ϵ	dielectric constant
μ_p	hole mobility	μ_n	electron mobility
ξ_p	hole electrochemical potential	ξ_n	electron electrochemical potential
J_p	hole current density	J_n	electron current density
j	j-th grid point in model	h_j	distance between grid points j and j+1

Note: Any variables with subscript j denotes value in lumped discrete model.

Lumped network models to approximate the characteristics of this coupled set of partial differential equations (PDEs) have been developed to describe the physical phenomena occurring in semiconductor materials. The work of Linvill [1] and that of Kani and Ohtsuki [2] takes this set of PDEs and differences them with respect to the space variable and arrives at a set of coupled ordinary differential equations (ODEs); they relate these ODEs to a nonlinear lumped RC network by developing a correspondence between the hole and electron concentrations and the electrostatic potential to the classical voltage and current in electrical networks. Another work based on approximating the distributed effect in semiconductor materials is that of Ghosh [3] in which he developed a model with multiple sections with each section being an Ebers-Moll model. Conceptually, these lumped network models are all the same, in that a lumped sectionalized model is used to approximate distributed physical phenomena in a semiconductor device.

Recent semiconductor device modeling efforts have emphasized the computer usage as the modeling tool to arrive at what is referred to as numerical models. The numerical efforts have also discretized the PDEs to arrive at a coupled set of ODEs (by discretizing only in space), or a coupled set of finite difference equations (FDEs) (by discretizing both space and time variables). Since the basic work originating numerical studies [4], more recent attempts have been concerned with developing numerical algorithms which are very efficient yet have sufficient accuracy in predicting the terminal characteristics of semiconductor devices [5-7].

Gummel and Poon [8] have developed a popular model based on the Ebers-Moll model. Their work modified the basic Ebers-Moll model by adding various elements of proper type and value such that the BJT transistor characteristics of all but the most difficult of circuit applications can be accomodated. However, the modeling of the internal physical phenomena has the advantage that it is readily extendable to all type situations from diodes to BJTs, JFETs, and MOSFETs, and conceptually to ICs. From an overall modeling point-of-view, by defining the doping process and semiconductor material parameters, the elements of the numerical model and/or lumped network model are directly defined. The necessity of having to match terminal characteristics to obtain the model parameters (as in the Gummel and Poon model) is avoided.

Previous lumped network modeling efforts have not had any concern relative to what may lead to a computationally efficient model. In contrast, the massive use of computers has inspired tremendous effort to arrive at extremely efficient numerical models. Two ideas motivating this study were (1) most circuit designers prefer to use a network model rather than a numerical model, and (2) most critical circuit analysis and design is done with the benefit of computer-aided-analysis-and-design (CAD) programs.

This paper reports on an investigation that uses the computationally efficient numerical models and generates likewise efficient lumped network models; that is, this paper reports on a model conversion process. What is new about this modeling effort is that it takes the numerical model that is used by the device designer to study the internal physical phenomena of devices and directly converts that model to a network model which the circuit designer can use immediately in his circuit design work. With this advanced modeling facility, the communications gap between the device and circuit designers is narrowed.

A further concern of this investigation is once these models are converted, how will they be incorporated to make CAD programs more efficient. An illustration of efficient CAD using these lumped-network/numerical models will be included. The numerical model of the internal device physics can be represented by a very desirable band-structured matrix. Through the use of partitioned matrices, CAD algorithms can take advantage of this band-structured matrix to lead to more efficient CAD programs.

NETWORK MODEL DEVELOPMENT

The computer generated model being presented has the same inclination as the modeling procedure of Kani and Ohtsuki [2]. There, as here, the idea was to arrive at a model made up of normal network elements; that is, a network containing resistances, capacitances, inductances, independent sources, and controlled or dependent sources. Any of these elements may be nonlinear; however, present day computer-aided-network-analysis programs will handle these with ease. Kani and Ohtsuki chose a network model containing only lumped RC elements, as shown in Fig. 1.

The overall approach in arriving at the proposed computer generated network model is the following. First, equations which were formulated because of their efficiency of numerical solution and which have demonstrated this efficiency in practice will be given. Secondly, rearrangement of these equations and identification of coefficients of the unknown variables will allow us to make an analogy between these mathematical discretized equations and equations resulting from networks. And finally, a lumped network model may be drawn from the set of carefully arranged equations; this lumped network model is guaranteed to lead to efficient computer solution.

The idea is to have a one-to-one relationship between the numerical model and the elements of the lumped network model. Thus with this matching of elements in the two models, the numerical model and the lumped network model will have identical topological characteristics.

The difference equations which will be used to generate a lumped network model are shown in Eqs. (7-9). Those equations result in a very desirable efficient computer solution.

Hole Continuity Equation:

$$\frac{\partial p_j}{\partial t} = \frac{2/q}{h_j+h_{j-1}} \left[q\mu_{p,j+1/2} \frac{p_j+p_{j+1}}{2} \left(\frac{v_{j+1}-v_j}{h_j} + \frac{d\xi_{p,j+1/2}}{dp} \frac{p_{j+1}-p_j}{h_j} \right) - q\mu_{p,j-1/2} \frac{p_j+p_{j-1}}{2} \left(\frac{v_j-v_{j-1}}{h_{j-1}} + \frac{d\xi_{p,j-1/2}}{dp} \frac{p_j-p_{j-1}}{h_{j-1}} \right) \right] - R_j \qquad (7)$$

Poisson's Equation:

$$\frac{2}{h_j+h_{j-1}} \left(\frac{v_{j+1}-v_j}{h_j} - \frac{v_j-v_{j-1}}{h_{j-1}} \right) = -\frac{q}{\epsilon} (N_j - n_j + p_j) \qquad (8)$$

Electron Continuity Equation:

$$\frac{\partial n_j}{\partial t} = -\frac{2/q}{h_j+h_{j-1}} \left[q\mu_{n,j+1/2} \frac{n_j+n_{j+1}}{2} \left(\frac{v_{j+1}-v_j}{h_j} - \frac{d\xi_{n,j+1/2}}{dn} \frac{n_{j+1}-n_j}{h_j} \right) - q\mu_{n,j-1/2} \frac{n_j+n_{j-1}}{2} \left(\frac{v_j-v_{j-1}}{h_{j-1}} - \frac{d\xi_{n,j-1/2}}{dn} \frac{n_j-n_{j-1}}{h_{j-1}} \right) \right] - R_j \qquad (9)$$

Now rearranging and rewriting these equations, to get the dimension of current in each term, and also taking the time derivative of Poisson's equation, Eqs. (10-13) below will be used directly to obtain the elements in the lumped network model. A good analogy is to treat all junctions as "potentials," and thus all elements will be conductances, capacitances, and voltage controlled current sources. It is not necessary to take this viewpoint, it just helps in the circuit and/or network analogy to Kirchhoff's current law. The cross-sectional area A is introduced into the equations at this point.

$$\begin{aligned} \frac{\partial}{\partial t} \left(q p_j \frac{h_j+h_{j-1}}{2} A \right) = {} & q\mu_{p,j+1/2} A \frac{p_j+p_{j+1}}{2h_j} (v_{j+1}-v_j) \\ & + q\mu_{p,j+1/2} A \frac{p_j+p_{j+1}}{2h_j} \frac{d\xi_{p,j+1/2}}{dp} (p_{j+1}-p_j) \\ & - q\mu_{p,j-1/2} A \frac{p_j+p_{j-1}}{2h_{j-1}} (v_j-v_{j-1}) \\ & - q\mu_{p,j-1/2} A \frac{p_j+p_{j-1}}{2h_{j-1}} \frac{d\xi_{p,j-1/2}}{dp} (p_j-p_{j-1}) - qR_j \frac{h_j+h_{j-1}}{2} A \end{aligned} \qquad (10)$$

$$\frac{\epsilon}{h_j}(v_{j+1}-v_j)A-\frac{\epsilon}{h_{j-1}}(v_j-v_{j-1})A=-q(N_j-n_j+p_j)\frac{h_j+h_{j-1}}{2}A \tag{11}$$

$$\frac{\epsilon}{h_j}(\dot{v}_{j+1}-\dot{v}_j)A-\frac{\epsilon}{h_j}(\dot{v}_j-\dot{v}_{j-1})A=+q\frac{h_j+h_{j-1}}{2}(\dot{n}_j-\dot{p}_j)A \tag{12}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left(qn_j\frac{h_j+h_{j-1}}{2}A\right)= & -q\mu_{n,j+1/2}A\frac{n_j+n_{j+1}}{2h_j}(v_{j+1}-v_j)\\ & +q\mu_{n,j+1/2}A\frac{h_j+h_{j+1}}{2h_j}\frac{d\xi_{n,j+1/2}}{dn}(n_{j+1}-n_j)\\ & +q\mu_{n,j-1/2}A\frac{h_j+h_{j-1}}{2h_{j-1}}(v_j-v_{j-1})\\ & -q\mu_{n,j-1/2}A\frac{h_j+h_{j-1}}{2h_{j-1}}\frac{d\xi_{n,j-1/2}}{dn}(n_j-n_{j-1})\\ & -qR_j\frac{h_j+h_{j-1}}{2}A\end{aligned} \tag{13}$$

By introducing the device cross-sectional area A, the following terms can be defined, having the proper units of charge, current, capacitance, and conductance.

$$C_j=\frac{\epsilon A}{h_j} \qquad Q_{c,j}=C_j\,(v_j-v_{j+1})$$

$$Q_{p,j}=qp_j\frac{(h_j+h_{j-1})}{2}A \qquad g_{mp,j}=q\mu_{p,j+1/2}\frac{p_j+p_{j+1}}{2h_j}A$$

$$I_{p,j+1/2}=A\,J_{p,j+1/2} \qquad G_{p,j}=q\mu_{p,j+1/2}\frac{p_j+p_{j+1}}{2h_j}\frac{d\xi_{p,j+1/2}}{dp}A$$

$$I_{r,j}=qR_j\frac{h_j+h_{j-1}}{2}A$$

Similar definitions may be made for the electron terms. The terms in Eqs. (10-13) have been purposefully separated so as to indicate the elements of the network model. Taking p_j, v_j, and n_j as "potentials," then it is obvious that the first term of Eq. (10) is a voltage controlled current source, the current being controlled by the voltage difference $(v_{j+1}-v_j)$ with the nonlinear proportionality parameter being $g_{mp,j}$. The second term in Eq. (10) represents a nonlinear "conductance" times the potential across it; the conductance being $G_{p,j}$. The last term in Eq. (10) represents a nonlinear recombination current flowing between the junctions p_j and n_j. Similar analogies may be made for the terms in the electron continuity equation. For Poisson's equation, a linear capacitance C_j can be defined between the j and j+1 voltage nodes. This term actually has dimensions of farads. The terms on the left-hand-side of Eqs. (10) and (13) appear also on the right-hand-side of Eq. (12); these terms are time derivatives of capacitive charges. The network model for these equations is shown in Fig. 2. From this model it can be seen that the total current is the sum of the hole and electron conduction currents plus the displacement current through the linear capacitance; thus the model agrees with Eqs. (6).

The proposed model is a mutually coupled network. One form of mutual coupling arises in the current sources in the hole and electron current branches being controlled by the voltage branches. The voltage controlled current sources are strikingly familiar to the g_mV in the hybrid-pi model, the common small-signal model for linear applications. Another form of mutual coupling is the current source represented by $(\dot{Q}_{p,j} - \dot{Q}_{n,j})$ in the potential branch; this current source is controlled by the current flow through the nonlinear capacitors connected from the p_j and n_j points to ground. The coupling element between the hole and electron points in the recombination current flow.

The network model being proposed here does not have a restriction to RC elements. In fact, the model being proposed will have controlled sources along with the RC elements. The reason for this difference (compared to all other network modeling efforts) is that here no attempt has been made to transform the PDEs such that all junction points in the model represent "node voltages." Instead there is a junction for p, a junction for v, and a junction for n. Each of these junctions represents a place for current summation (which must equal zero value), with one junction for hole current, a second for displacement current, and the third for electron current, respectively.

The major difference in this proposed network model and others is that it has been developed solely from the point-of-view that computer aids will be used to arrive at a solution. Since the mathematical formulation of the difference equations was based mainly on its very efficient computer analysis allowed, and the network model was developed solely for making more efficient the inevitable computer aided network analysis, then the model being proposed can be considered as "a computer generated semiconductor network model."

Another "concession" that can be allowed in this model is that of scaling. The scaled equations remove all unnecessary constants such as q and ϵ and significantly reduces the multiplicative operations involved, as these two variables enter numerous times only as multiplication factors. The only "problem" arises at the interface of the semiconductor device and the surrounding circuit; this problem is very minor because the voltage scaling factor must be introduced again into the sets of boundary equations and into terms in the circuit matrix.

CIRCUIT CONSIDERATIONS

Use of Model in Circuit

After obtaining this network model of the semiconductor device, the question arises as to the use of the model in some surrounding circuit containing linear elements and maybe more nonlinear elements which also have lumped network models. Each network model when transformed into its describing mathematical equations is a well-ordered, sparse, band-structured, efficiently solvable nonsingular matrix relation. Each model produces a terminal voltage-current (v-i) relationship, the same as any other electrical element in a circuit. Thus we can write node equations, loop equations, or even use the tableau approach [9]. The resulting matrix structure for the complete circuit will be a partitioned matrix, with particular emphasis on the semiconductor device matrices being kept in their well-ordered, sparse, and efficiently solvable form. The equations for the surrounding circuit (all linear) are ordered such that they conform to this partitioned matrix form.

To solve this partitioned matrix equation, when ordering for the matrix factorization one should be careful to move matrices rather than individual elements in the matrix. Individual elements for the linear terms can be used as "pivot" elements provided that these terms do not directly couple one of the device terminal voltage or current unknown variables. Thus whenever one of the device variables must be used as a "pivot" element, the complete device matrix structure becomes the diagonal pivot element; that complete matrix structure is factored, and then the reduction continues at the next variable.

An Example

Fig. 3 shows a resistor-diode circuit. The use of partitioned matrices is to be used to solve this circuit problem containing the nonlinear diode imbedded in the linear circuit of the voltage source and resistor. Even though this example is simple, the concepts invoked are easily extended to any size circuit.

The equations involved are

$$v_s - v_d = R\,i \qquad (14)$$

$$i = i_d \qquad (15)$$

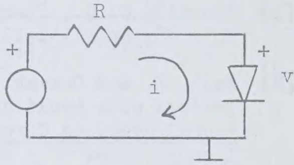


Fig. 3

These equations when solved will give a matrix structure as follows (assuming that Newton's method is being used):

$$\begin{array}{cc} v_d & i \\ \left[\begin{array}{c} \begin{bmatrix} & D & \end{bmatrix} \\ 0\ldots0\,{-1} \end{array}\right. & \left.\begin{array}{c} +1\\ 0\\ 0\\ -R \end{array}\right] \end{array} \qquad \begin{array}{l} f_1 \text{ (diode equations)} \\ f_2 = v_s - v_d - i\,R \end{array} \qquad (16)$$

where D is the matrix describing the internal physical phenomena which relates the terminal current-voltage i_d-v_d relationship. Proper handling of this matrix leads to efficient solution and does not introduce a lot of extra computation and storage. The matrix D is a well-ordered, sparse, band-structured, nonsingular matrix. The complete circuit matrix can be written as a partitioned matrix as

$$\begin{bmatrix} D & C \\ B & A \end{bmatrix} \qquad (17)$$

where

$$A = [-R]. \qquad C = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (18)$$

$$B = [0\ .\ .\ .\ 0\ {-1}],$$

The -1 in B corresponds to the coefficient of v_d, and the 1 in C corresponds to the diode current equation in the diode equations. Now reducing this partitioned matrix to LU factorization results in the following (using the standard partitioned matrix procedure)

$$L = \begin{bmatrix} D & 0 \\ B & A-BD^{-1}C \end{bmatrix}, \qquad U = \begin{bmatrix} I & D^{-1}C \\ 0 & I \end{bmatrix} \qquad (19)$$

Thus when a submatrix solution is required (denoted by the inverse notation) only the efficiently solvable matrix D is used, and the well-ordered, sparse, and band-structured matrix D is not destroyed.

CONCLUSION

In this paper a model conversion method has been proposed for generating a lumped network model suitable for circuits in high-performance large-signal operation. Yet the network model which was conceptually "computer-generated" is very efficient and retains the computational efficiency of the well-investigated numerical models. One efficient use of the resulting well-ordered, sparse, band-structured, efficiently solvable nonsingular matrix in a surrounding circuit matrix has been shown to be feasible by the use of partitioned matrix concepts. An example was presented to highlight this partitioned matrix procedure.

REFERENCES

[1] Linvill, J. G., "Lumped Models of Transistors and Diodes," Proc. IRE, Vol. 46, pp. 1141-1152, June 1958.

[2] Kani, K. and Ohtsuki, T., "A Unified Modeling Scheme for Semiconductor Devices with Applications of State Variable Analysis," IEEE International Symp. on Circuit Theory, Dec. 1968, and IEEE Trans. Circuit Theory, 1969.

[3] Ghosh, H. N., "A Distributed Model of the Junction Transistor and its Application in the Prediction of the Emitter-Base Diode Characteristic, Base Impedance, and Pulse Response of the Device," IEEE Trans. Electron Devices, Vol. ED-12, No. 10, pp. 513-531, Oct. 1965.

[4] Gummel, H. K. "A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations," IEEE Trans. Electron Devices, Vol. ED-11, pp. 455-465, Oct. 1964.

[5] DeMari, A., "An Accurate Numerical One-Dimensional Solution of the p-n Junction Under Arbitrary Transient Conditions," Solid-State Electronics, Vol. 11, No. 11, pp. 1021-1053, Nov. 1968.

[6] Hachtel, Gary D., "On the Application of Stone's 'Semi-Direct' Elimination Method to the Solution of the Poisson's and Continuity Equations Describing 2-Dimensional Semiconductors," Digest Record of 1969 Joint Conf. on Math. and Comp. Aids to Design, Anaheim, California, pp. 356, Oct. 1969.

[7] McAfee, L. C., "The Optimization of Numerical Models of Semiconductor Devices," Eighth Annual Allerton Conf., pp. 625-634, Oct. 1970.

[8] Gummel, H. K. and Poon, H. C., "An Integral Charge Control Model of Bipolar Transistor," Bell System Technical Journal, Vol. 49, No. 5, pp. 827-852, May-June 1970.

[9] Hachtel, Brayton, and Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 101-113, Jan. 1971.

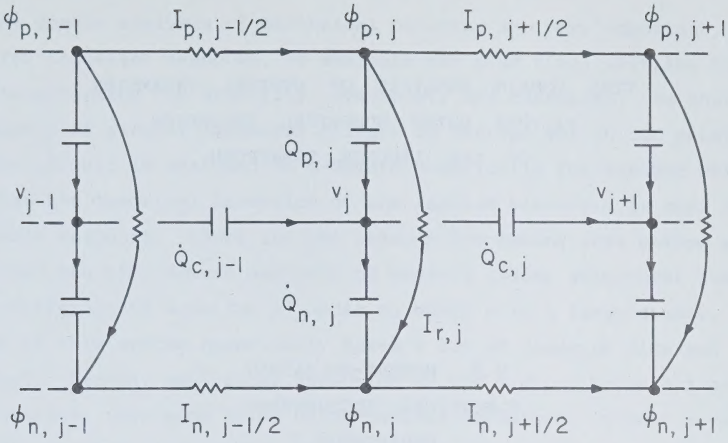

$\phi_{p,j-1}$
$I_{p,j-1/2}$
$\phi_{p,j}$
$I_{p,j+1/2}$
$\phi_{p,j+1}$
$\dot{Q}_{p,j}$
v_{j-1}
v_j
v_{j+1}
$\dot{Q}_{c,j-1}$
$\dot{Q}_{c,j}$
$\dot{Q}_{n,j}$
$I_{r,j}$
$\phi_{n,j-1}$
$I_{n,j-1/2}$
$\phi_{n,j}$
$I_{n,j+1/2}$
$\phi_{n,j+1}$

FIG. 1 KANI AND OHTSUKI NETWORK MODEL.

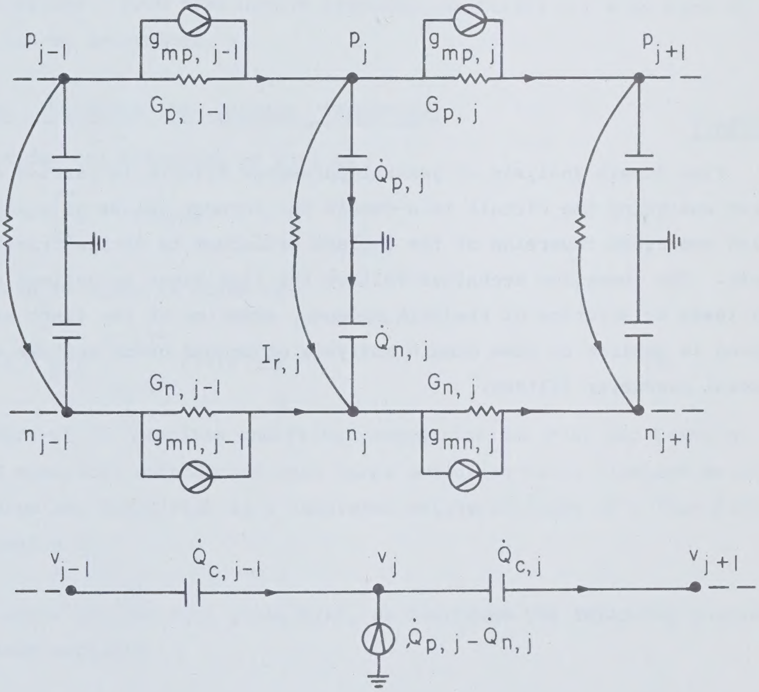

p_{j-1}
$g_{mp,j-1}$
p_j
$g_{mp,j}$
p_{j+1}
$G_{p,j-1}$
$G_{p,j}$
$\dot{Q}_{p,j}$
$\dot{Q}_{n,j}$
$I_{r,j}$
$G_{n,j-1}$
$G_{n,j}$
n_{j-1}
$g_{mn,j-1}$
n_j
$g_{mn,j}$
n_{j+1}
v_{j-1}
$\dot{Q}_{c,j-1}$
v_j
$\dot{Q}_{c,j}$
v_{j+1}
$\dot{Q}_{p,j} - \dot{Q}_{n,j}$

FIG. 2 COMPUTER GENERATED COMPUTATIONALLY EFFICIENT NETWORK MODEL.

TIME DOMAIN ANALYSIS OF GENERAL PARAMETER FILTERS USING NUMERICAL INVERSION OF THE LAPLACE TRANSFORM

M.E. MOKARI-BOLHASSAN
ELECTRICAL ENGINEERING
DEPARTMENT
PAHLAVI UNIVERSITY
SHIRAZ, IRAN

ABSTRACT

Time domain analysis of general parameter filters is carried out by first analyzing the circuit in s-domain for integer values of s and then using numerical inversion of the laplace transform to obtain transient response. The inversion technique follows the line given by Bellman et.al. 1 and leads to solution of Fredhalm intagrol equation of the first kind. The method is applied to time domain analysis of second order systems and to general parameter filters.

INTRODUCTION

Time domain analysis of electrical networks are very important, especially for transient response, to evaluate the rise time, settling time, etc. mainly to determine the stability, overshort, and bandwidth. In this paper the analysis of general parameter filters is carried out in two phases. First the circuit is analysed in s-domain numerically for integer values of s and then the numerical inversion of the laplace transform is used to obtain time domain response. There are two reasons for making this method attractive. One, the time domain analysis of general ladder structures leads to integro-differential equation of order n, where n is a large number. Solution of this system numerically takes a lot of computer time and is not economical. Second, analytical inversion of the laplace transform or any other numerical inversion (i.e. using residue technique) is much more complicated and sometimes impossible. The proper method would be to evaluate s-domain response numerically in minimum required number of points (perferably for real s) and using the inversion technique discussed below to obtain time domain response. From time domain response, stability and rise time of the network can be determined.

NUMERICAL INVERSION OF LAPLACE TRANSFORMS

Consider the transform of f(t) as

$$F(s) = \int_{0}^{\infty} e^{-st} f(t)dt \qquad (1)$$

The inverse relation is given by

$$f(t) = \frac{1}{2\pi j} \int_{\sigma_1 - j\infty}^{\sigma_1 + j\infty} F(s)e^{st} ds \qquad (2)$$

Application of (2) requires analytical expression for F(s) and leads to complicated numerical methods and very large computing time. Instead we pose the problem as: Given F(s) as a tabulated values in terms of s find f(t) for corresponding t.

To solve (1) for f(t) given F(s), we intriduce the following change of independent variable

$$e^{-st} = x \qquad (3)$$

Equation (1) reduces to

$$\int_0^1 x^{s-1} u(\ln x)dx = F(s) \tag{4}$$

To approximate the left hand side of (4) by a finite sum the following considerations are first given.

Consider the approximation

$$\int_{-1}^1 f(r)dr \simeq \sum_{i=1}^N w_i f(r_i) \tag{5}$$

where w_i are positive weighting functions and k_i are properly picked points. To evaluate w_i and r_i we need to make certain requirements on (5). One way is to require that (5) be exact for any polynomial of degree less than or equal to 2N-1. It gives 2N conditions to evaluate w_i, r_i, i=1, ..., N. This is called Gaussa's method. For this f(r) has to be approximated by

$$f(r) = \sum_{k=0}^N a_k r^k \tag{6}$$

For completeness of the class of the functions the functions are required to be orthogonal. Therefore polynomials in (6) are not complete set, hence we take legendre polynomials and approximate f(r) by

$$f(r) = \sum_{k=0}^N b_k p_k(r) \tag{7}$$

where $p_k(r)$ is orthogonal lengendre polynomial of order k. To find b_k's it is required to minimize

$$J = \int_{-1}^1 \left[f(r) - \sum_{k=0}^N b_k p_k(r)\right]^2 dr \tag{8}$$

Setting $\nabla_b J = 0$ gives

$$b_\ell = \frac{2\ell+1}{2} \int_{-1}^1 f(r) p_\ell(r) dr \tag{9}$$

Referring back to equation (5) we need to find w_i and r_i's. By Gauss's method these parameter are given as [1,2,3].

$$r_i,\ i=1,2,\ \ldots,\ N \text{ are zeros of } P_N(r) \tag{10}$$

and

$$w_i = \int_{-1}^{1} \frac{P_N(r)}{(r-r_i)P_N'(r_i)} dr \qquad (11)$$

where $P_N(r)$ is legendre polynomial of degree N and prime indicates differentiation. The constants r_i and w_i are fixed for a given N and can be evaluated once for further use [1,3].

Applying the above argument to equation (4) gives

$$\int_0^1 x^{s-1} u(-\ln x)dx = F(s)$$

or

$$\int_0^1 x^{s-1} g(x)dx = F(s) \qquad (12)$$

which is approximated by

$$\sum_{i=1}^{N} w_i x_i^{s-1} g(x_i) = F(s) \qquad (13)$$

Letting s assume N different values, say s=1,2,...,N, yields a linear system of N equations in N unknowns $g(x_i)$, i=1,2,...,N. Or

$$\sum_{i=1}^{N} w_i x_i^{k} g(x_i) = F(k+1),\ k=0,1,\dots,N-1 \qquad (14)$$

where $w_i's$ are the weights indicated above and x_i are zeros of the shifted legendre polynomials [1,2,3]. Solving (13) for $g(x_i)$ gives the desired function $f(t_i)$ at points where

$$t_i = -\ln x_i \qquad (15)$$

For linear systems, it is also possible to obtain an explicit relation for $g(x_i)$ [3].

The largest t_i obtained in (14) for different N is limited by 5.115 sec. for N=15. If f(t) is needed for larger values of t we use time scaling of laplace transform, namely

$$L\ f(at) = \frac{F(\frac{s}{a})}{a} \qquad (16)$$

Picking proper a one can obtain the response for large t's and at t's which is interpolated with points obtained for a=1 or different a.

Finally the method discussed here is not the only method for inversion of the laplace transform. One may use any sets of orthogonal polynomials. In refference [4] laguerre polynomials are used which required some cimplex domain calculation. The above method is carried out in real numbers and is also simple to implement. In the following the method is applied to time domain analysis of ladder filters.

EXAMPLE

Numerical inversion of the laplace transform has been applied to time domain analysis of both second order systems and general parameter filters. For this we have evaluated step response of the corresponding systems.

1. Second order systems

The general equation for this class is taken as

$$\frac{V_o(s)}{V_i(s)} = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} \tag{17}$$

where ξ is the damping ratio and ω_n is the undamped natural frequency. Figure 1,2, and 3 the graph of numerically calculated step response of this system for under-damped, critically damped, and overdamped cases. In Figure 4 the effect of N on the accuracy ("reasonableness") of the solution is shown. Theoretically as N increases the solution approaches the true solution. For N=15 the solution has become unstable. The only justification for this is the round-off and accumulated error in the computer. In IBM 1130 the result of Figure 4 is obtained, in other computers this result will be different. The analytical solution and the numerical result are in a good agreement for N=11. In the following example N=11 was used.

2. General parameter filters

The step response of ladder structure filters are obtained for both lossless and lossy elements. A program is written to analyse these filters in s-domain and then using the numerical inversion technique the results are obtained. Step response of low-pass filter of Figure 5 is shown in Figure 6. The calculated points are shown along with curve fit. Figure 7 shows the response of both lossy and lossless lowpass filters. Figure 9 shows the step response of the band-pass filter of Figure 8. Figure 10 shows the lossy response. Figure 12 and 13 shows the step response of the band-elimination filter of Figure 11. Finally, the step-response of high-pass filter of Figure 14 is shown in Figure 15 and the lossy element response is shown in Figure 16. It should be mentioned that the responses are all acceptable and some of the results were checked experimentally and the results are in agreement with the numerically obtained values.

CONCLUSION

In this paper the application of the numerical inversion of the Laplace transform to time domain analysis of general parameter filters (ladder structures) is studied. It is observed that this method is most useful and economical to evaluate approximate transient response. The method is best suited for large systems since both s-domain analysis and the inversion are carried out numerically, only at a few points. An optimal utilization of s-domain information leads to an economical analysis method. Different polynomials could be used in the approximation of the problem which may be useful in different cases and ill-conditioned problems [4].

REFERENCES

1 R.E. Bellman, R.E. Kalaba, and J.A. Kockitt, "Numerical Inversion of the Laplace Transform and Application," American Elsekier Publishing Co., New York, 1966.

2 R.E. Bellman, H.H. Kagiwada, and R.E. Kalaba, "Numerical Inversion of Laplace Transform and Some Inverse Problems in Radiative Transfer," J. of the Atmospheric Science, Vol. 23, No. 5, pp. 555-559, Sep. 1966.

3 M. Pakbaz and M.E. Mokari-Bolhassan, "Time Domain Analysis of Ladder Filters," Internal Report, EE Dept., Pahlavi University, Shiraz, June 1971.

4 R. Piessens and M. Branders, "Numerical Inversion of Laplace Transform Using Generalized Laguerre Polynomials," Proc. IEE, Vol. 118, No. 10, Oct. 1971.

FIGURE CAPTIONS

Figure 1. Unit step response of a second order system (under-damped)
Figure 2. Unit step response of a second order system (critically damped)
Figure 3. Unit step response of a second order system (over-damped)
Figure 4. Effect of N on the accuracy of the solution in IBM 1130
Figure 5. Low-pass filter
Figure 6. Unit step response of the low-pass filter
Figure 7. Unit step response of the low-pass filter (lossy)
Figure 8. Band-pass filter
Figure 9. Unit step response of the band-pass filter
Figure 10. Unit step response of the band-pass filter (lossy)
Figure 11. Band-elimination filter
Figure 12. Unit step response of the band-elimination filter
Figure 13. Unit step response of the band-elimination filter (lossy)
Figure 14. High-pass filter
Figure 15. Unit step response of the high-pass filter
Figure 16. Unit step response of the high-pass filter (lossy).

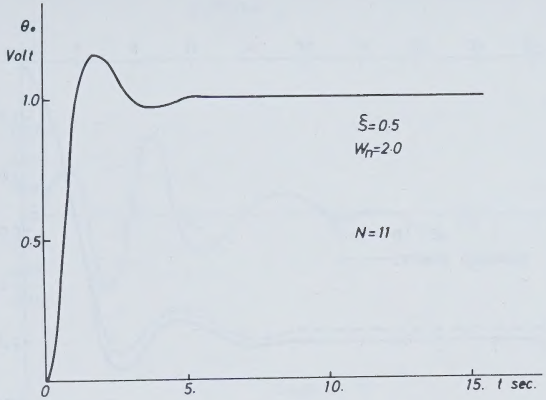


Figure 1

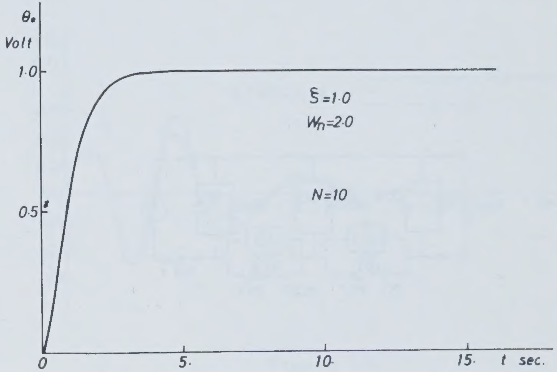


Figure 2

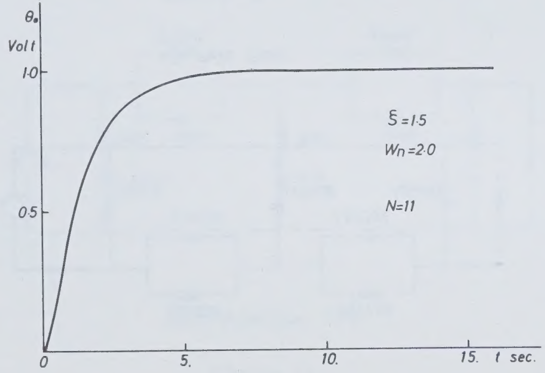


Figure 3

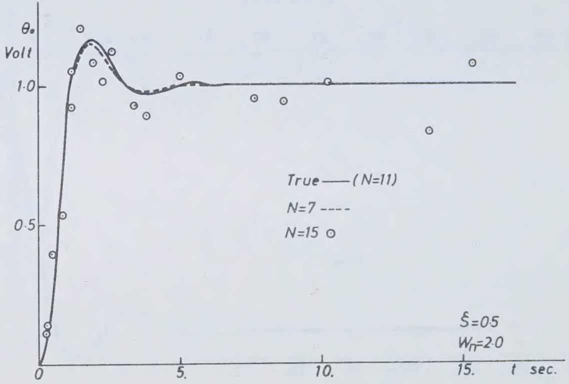
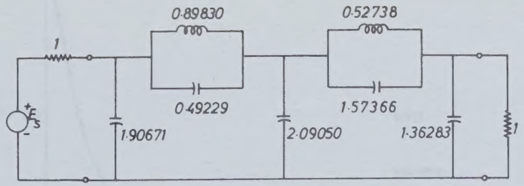


Figure 4


0.89830
0.52738
1
0.49229
1.57366
E_s
1.90671
2.09050
1.36283
1

Low-Pass Filter

Figure 5

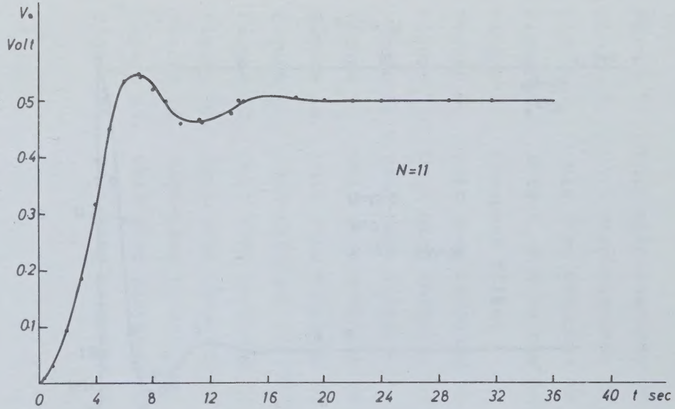


Figure 6

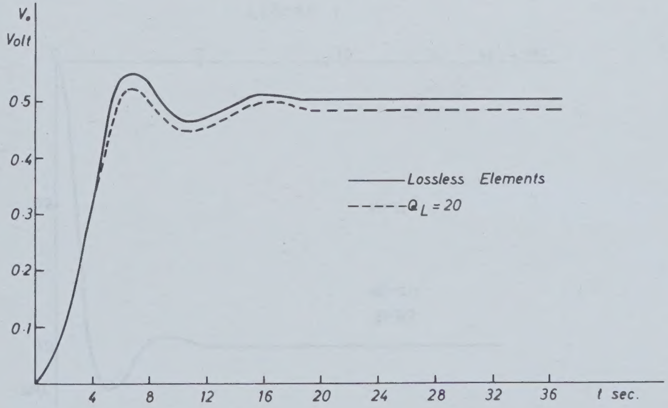


Figure 7

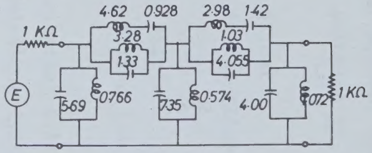

4.62
0.928
2.98
1.42
1 KΩ
3.28
1.03
1.33
4.055
E
5.69
0.766
7.35
0.574
4.00
0.72
1 KΩ

Figure 8

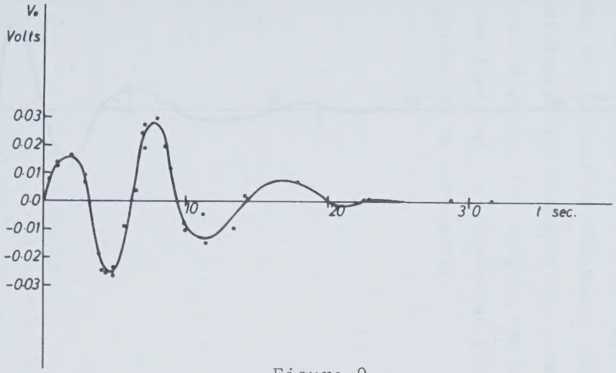


Figure 9

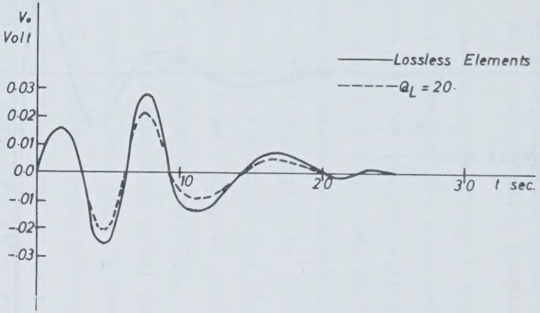
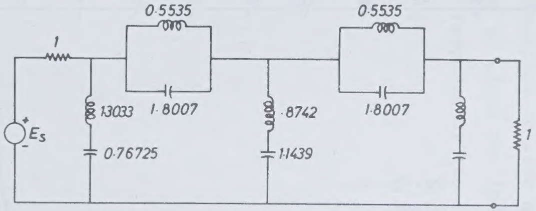


Figure 10


1
0.5535
1.8007
1.3033
0.76725
.8742
1.1439
0.5535
1.8007
1
E_s

Band Elimination Filter

Figure 11

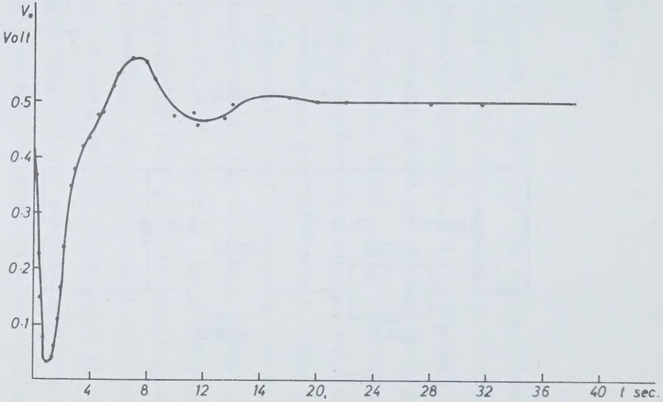


Figure 12

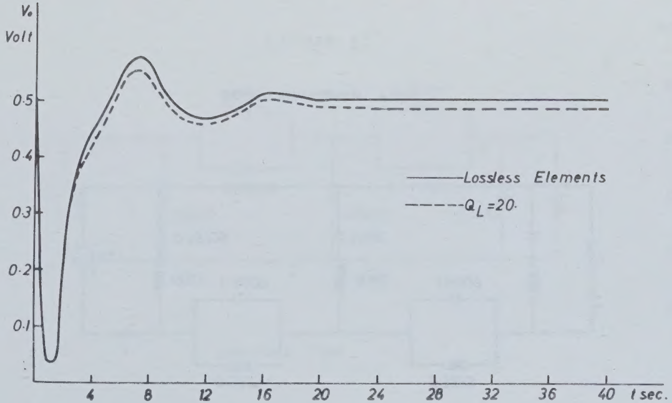
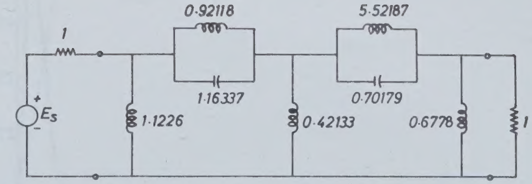


Figure 13


1
0·92118
1·16337
5·52187
0·70179
E_s
+
-
1·1226
0·42133
0·6778
1

High-Pass Filter

Figure 14

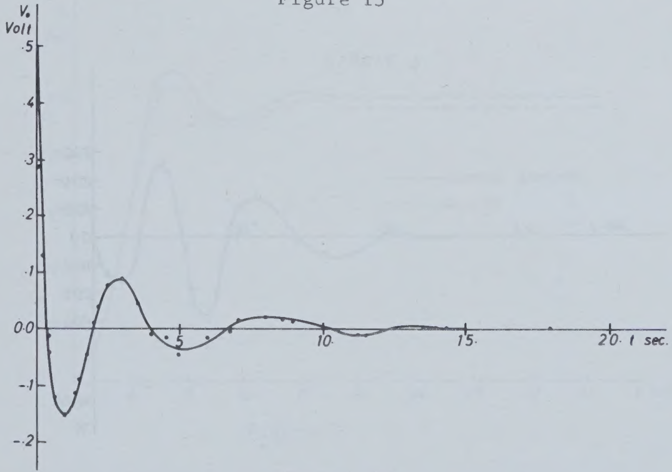


Figure 15

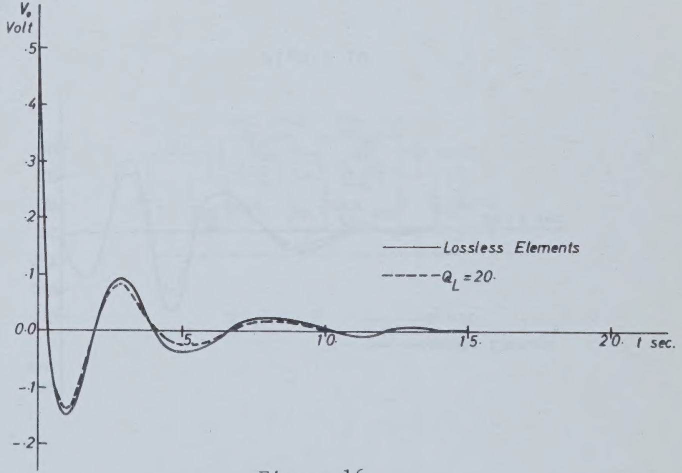


Figure 16

THE SIMULATION OF CONTINUOUS TRANSFER FUNCTIONS BY USE OF PCAP

SIDNEY M. SCARBOROUGH
Department of Electrical Engineering
Mississippi State University
Mississippi State, Mississippi

The Princeton Circuit Analysis Program (PCAP) and the Univac 1106 digital computer have been used successfully for the simulation of continuous, linear systems with known transfer functions. The procedure is easy to use; it makes use of the PCAP language which is already well known by the undergraduate students in many electrical engineering curricula; the coefficients and gain constant of the transfer function are easily varied; both transient and sinusoidal responses can be studied. The first step is to draw a diagram containing integrators and summers to realize the given transfer function. Next, the obvious step would be to use the operational amplifier model contained in PCAP to realize the integrators and summers. When this is done, the resulting PCAP program produces an underflow (using single precision arithmetic), except in small systems containing only three or four op amps. A more useful alternative is to realize the summers with voltage controlled, voltage sources having more than one controlling branch and to realize the integrators with voltage controlled current sources with capacitors being the passive elements in the current-source branches. After the open loop response has been determined and possibly the coefficients modified to give a desired open loop response, the program can be modified to close the loop and thus the transient and the sinusoidal steady state response of the closed loop system can be determined.

MIXED DOMAIN TRANSIENT ANALYSIS OF LARGE NON-LINEAR NETWORKS

C. B. FRYE, JR. and M. J. APFELBAUM
Applied Systems Modeling Department
Communication Systems Division
GTE Sylvania Inc.
Needham Heights, Massachusetts, 02194

ABSTRACT

A new approach to the transient analysis of large non-linear networks requires the capability of performing multiple convolutions in a step-by-step fashion during the execution of a time domain program. This paper describes the development of such a technique. In particular, since none of the existing CAD programs currently available had this capability, we elected to extend the SCEPTRE program, thereby providing it with a multiple convolution capability.

I. INTRODUCTION

BACKGROUND

In engineering practice we encounter networks of 10^2 to 10^4 elements on which transient analysis is desired. The networks contain a moderate number of non-linearities, something in the order of 10 to 20 devices, and a relatively small number of penetration points, also of the same order.

A need exists for the time domain solution of such networks. However, their size precludes transient solution in the conventional manner using even the most sophisticated CAD programs available. Fortunately, the networks encountered possess characteristics which suggest an analytic approach. They will, in general, allow one to consider the majority of elements as a large linear subnetwork where the following two conditions exist:

(a) The remaining elements form another subnetwork of moderate size. By moderate size we mean something less than the maximum allowed by SCEPTRE (1). We assume that all of the non-linearities are placed in this subnetwork. We also assume that all of the forcing functions (penetration points) are placed in this subnetwork.

(b) The two partitioned subnetworks are linked together with a small number of interconnecting lines. The intent is to represent the large linear subnetwork across this set of interconnecting lines. Only a small number of interconnecting lines is permitted, because the number of combinations, C, necessary to characterize the interaction between N lines, referenced to but excluding the datum line, is given by $C = N(N+1)/2$. Thus, for example, a 12 line interface would need 12 x 13 = 156 elements (at two elements per combination) for the interface, leaving approximately one-half, assuming a maximum of 300 elements, for the remaining non-linear circuitry.

CONVOLUTION ANALYSIS APPROACH

The convolution analysis approach, i.e., the mixed domain technique, adopted and used by GTE Sylvania for this type of problem consists of the following major steps:

(a) Partition the network into two parts (see Figure 1-1).

(b) Use a frequency domain program capable of solving a large linear network to determine the driving-point and transfer immittance functions, $H(\omega)$, at the interface terminals.

(c) Find the equivalent impulse response functions, $h(t)$, in the time domain by using Fourier inversion.

(d) Model these impulse response functions as either impedances or admittances. The impedance (admittance) model assumes that a removed portion of an existing circuit is characterized at the interface by a Thevenin voltage source (Norton current source) and a driving-point impedance (admittance).

(e) Describe these time domain functions in a way acceptable to SCEPTRE.

(f) Solve the remaining moderate-sized subnetwork containing the nonlinearities and include these impulse response functions using SCEPTRE with its convolution capability.

The first step is accomplished manually at the discretion of the user. The second step may be accomplished using any auxilliary AC analysis program which is capable of providing the desired immittance functions. TRAFFIC(2) is one such program. The third step is done using an off-line program, FOUR2(3), which processes the AC outputs and supplies the inverse Fourier Transforms. An existing program FOUR1(3), capable of performing a direct Fourier Transform using non-uniform steps, was modified and made available for the inverse Fourier processing.

The ability to do the all-important steps d, e, and f is the key to this method of analysis. The problem of modeling the time functions and processing them in synchronism with SCEPTRE has been solved and is the subject of this paper. This was done by observing that these functions form the impulse response matrix for the removed circuitry as seen by the original network across the appropriate interface terminals. Therefore, in the impedance mode, the time function of the voltage across any pair of interface terminals can be obtained by convolving the appropriate impedance function with the time function describing the current flowing through the two terminals. Similarly, in the admittance mode, the current flowing through the terminal can be obtained by convolving the voltage function, across the terminals, with the appropriate admittance function.

SCEPTRE calculates currents and voltages, for a given configuration, as functions of time. Therefore, in order to calculate present values, it was necessary that the convolution routine be written so that, in the impedance mode (admittance mode), it uses past history of the current (voltage) to estimate the present value of current (voltage), which in turn will yield the required terminal-to-terminal voltage (current). It then updates the current and voltage tables and repeats the procedures for the next time step.

II. THEORY

MATHEMATICAL FOUNDATION

GTE Sylvania's approach to the solution of the class of non-linear problems which contain a major linear subnetwork relies on the exploitation and extension of known techniques which hold for linear systems.(4) For example, use is made of the superposition principle by assuming that the total response of a large linear network, defined at the interface, is the sum of all the individual line-to-line contributions.

Furthermore, we employ the ideas embodied in the two fundamental procedures to obtain the output time response associated with an externally driven linear system. The two procedures are given below.

Convolution Given an excitation F(t), determine the system transfer function H (ω) and find the impulse response of the network:

$$H(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)\ e^{j\omega t}\ d\omega \tag{2-1}$$

Then, the output response is:

$$R(t) = \int_{-\infty}^{t} H(t-\tau)F(\tau)d\tau \tag{2-2}$$

Spectral Multiplication Given an excitation F(t), determine its equivalent spectral representation:

$$F(\omega) = \int_{-\infty}^{\infty} F(t)e^{-j\omega\ t}\ dt \tag{2-3}$$

Determine the system transfer function H(ω). Then, the spectrum of the output is given by:

$$R(\omega) = H(\omega)\cdot F(\omega), \tag{2-4}$$

and, by Fourier Inversion, the output time response is:

$$R(t) = \frac{1}{2\pi}\int_{-\infty}^{t} R(\omega)e^{j\omega t}\ d\omega \tag{2-5}$$

However, we must deviate from these two processes because of three specific differences in our problems: a. Our networks are, in general, non-linear, b. We do not have available the system's over-all transfer function and, consequently, we cannot use the Convolution approach directly, and c. We do not have the input spectral representation, F(ω), as required by the Spectral Multiplication approach, at the particular interface in question.

The non-linear aspect is handled by SCEPTRE automatically through the use of its numerical integration capability. The second and third differences are resolved by using a combination of Fourier inversion and convolution relative to the interface rather than at the over-all system level. Fourier Inversion is performed off-line to convert the frequency domain transfer functions to time domain impulse response functions. Individual convolutions, as described in the next subsection, are performed between each such impulse response function and its respective voltage or current counterpart within SCEPTRE.

INTEGRATION ROUTINE

In general the convolution integral for continuous functions is given by:

$$R(t) = \int_{0}^{t} H(t-\tau)F(\tau)\,d\tau, \quad \text{where } F(\tau) = 0 \text{ for } \tau < 0. \tag{2-6}$$

For a given t and a given subdivision of the τ axis, defined by:

$$T_o = 0 < T_1 < \ . \ . \ . \ T_{N-1} < T_N = t,$$

the integral in equation 2-6 becomes:

$$R(t) = \sum_{i=o}^{N-1} S_i, \text{ where } S_i = \int_{T_i}^{T_{i+1}} H(t-\tau)F(\tau)d\tau \tag{2-7}$$

We assume the H and F functions are given in tabular form and that the data points are correct at the sample times involved. We make use of these sample times in our selection of the subdivisions along the τ axis by considering these subdivision breakpoints to be the ordered set of points obtained by interleaving the two sets of time points which define the H and F functions, respectively.

We also assume that the two time functions are representable between consecutive time points as straight line segments through these data points. That is,

$$\left.\begin{array}{l} F(\tau) = A_i\,(\tau - T_i) + B_i \\ \text{and} \\ H(T_N - \tau) = C_i\,(\tau - T_i) + D_i \end{array}\right\} \quad T_i \leq \tau \leq T_{i+1}, \tag{2-8}$$

where

$$\left.\begin{array}{ll} A_i = \dfrac{F(T_{i+1}) - F(T_i)}{T_{i+1} - T_i} \ ; & B_i = F(T_i) \\ \\ C_i = \dfrac{H(T_N - T_{i+1}) - H(T_N - T_i)}{T_{i+1} - T_i} \ ; & D_i = H(T_N - T_i) \end{array}\right\} \tag{2-9}$$

The ordered set of interleaved time points define the non-zero subintervals which contribute to the over-all integration. The contribution from the ith general subinterval is:

$$S_i = \int_{T_i}^{T_{i+1}} H(T_N - \tau)F(\tau)d\tau$$

$$S_i = \int_{T_i}^{T_{i+1}} \left[C_i\,(\tau - T_i) + D_i\right] \quad \left[A_i\,(\tau - T_i) + B_i\right] \ d\tau$$

$$S_i = \int_{T_i}^{T_{i+1}} \left[A_i C_i (\tau - T_i)^2 + (A_i D_i + B_i C_i)(\tau - T_i) + B_i D_i \right] d\tau \qquad (2\text{-}10)$$

The integrand associated with this subinterval is seen to be a quadratic expression resulting from the product of the two straight line segments characterizing the two functions over the interval. The exact analytic integral of this equation is:

$$S_i = \left(\frac{A_i C_i}{3}\right)\left(T_{i+1} - T_i\right)^3 + \left(\frac{A_i D_i + B_i C_i}{2}\right)\left(T_{i+1} - T_i\right)^2 + B_i D_i \left(T_{i+1} - T_i\right) \qquad (2\text{-}11)$$

Let

$$\Delta T_i = T_{i+1} - T_i \, .$$

Then after substituting equation (2-9) into (2-11) and simplifying we get:

$$S_i = F(T_i)\left[\frac{H(T_N - T_{i+1}) + 2H(T_N - T_i)}{6}\right]\Delta T_i + F(T_{i+1})\left[\frac{2H(T_N - T_{i+1}) + H(T_N - T_i)}{6}\right]\Delta T_i \qquad (2\text{-}12)$$

Equation (2-12) describes the general form of the convolution integral over a subinterval in terms of the width of the subinterval and the four data points defining the two functions at each end point of the subinterval.

Two salient features of this integration scheme are:

a. It introduces no new time points. That is, it uses only subintervals defined by known time points.

b. It introduces no error of its own. This is the exact integral based on the straight line assumption, and there is no approximation to an integral involved. (Granted, the original data points may be in error and the straight line assumption is an approximation).

IMPEDANCE MODEL

If we regard the H function as an impedance function between two nodes, A and B, then the voltage across A-B and the current through A-B are related by the convolution integral given in equation (2-6). That is, the voltage at a particular time T_N is:

$$V_{AB}(T_N) = \int_0^{T_N} Z_{AB}(T_N - \tau) I(\tau) d\tau \qquad (2\text{-}13)$$

However, equation (2-13) cannot be solved directly as it is written, because the present value of current, $I(T_N)$, is not known. Examination of equation (2-13) indicates that the only term containing $I(T_N)$ is the n^{th} subinterval. Rewriting the equation in order to separate this term yields:

$$V_{AB}(T_N) = V1 + V2, \qquad (2\text{-}14)$$

where

$$V1 = \int_0^{T_{N-1}} Z_{AB}(T_N - \tau) I(\tau) d\tau \qquad (2\text{-}15)$$

and

$$V2 = \int_{T_{N-1}}^{T_N} Z_{AB}(T_N - \tau) I(\tau) d\tau \tag{2-16}$$

The V1 term involves only previously calculated values of current and can be represented as a variable battery, E_1 The V2 term, when expanded using equation (2-12), becomes:

$$V_2 = I(T_{N-1}) \left[\frac{H(T_N - T_N) + 2H(T_N - T_{N-1})}{6} \right] \Delta T_{N+1} + I(T_N) \left[\frac{2H(T_N - T_N) + H(T_N - T_{N-1})}{6} \right] \Delta T_N \tag{2-17}$$

The terms comprising V_2 can be considered to result from a series combination of another variable battery, E_2 , and a variable resistor, R.

Thus,

$$V_2 = E_2 + I(T_N) R. \tag{2-18}$$

And, letting $E = E_1 + E_2$:

$$V_{AB}(T_N) = E + I\ (T_N) R. \tag{2-19}$$

SCEPTRE can handle equation (2-19). At each step it evaluates E and R, solves for $I(T_N)$, and then calculates $V_{AB}(T_N)$. See Figure 2-1.

ADMITTANCE MODEL

If we regard the H function as an admittance function, then the current at a particular time T_N is:

$$I_{AB}(T_N) = \int_{0}^{T_N} Y_{AB}(T_N - \tau) V(\tau) d\tau \tag{2-20}$$

Similar to the procedure of the paragraph entitled "Impedance Model", we separate this integral into two terms

$$I_{AB}(T_N) = I_1 + I_2, \tag{2-21}$$

where

$$I_1 = \int_{0}^{T_{N-1}} Y_{AB}(T_N - \tau) V(\tau) d\tau \tag{2-22}$$

$$I_2 = \int_{T_{N-1}}^{T_N} Y_{AB}(T_N - \tau) V(\tau) d\tau \tag{2-23}$$

I_1 is represented as a variable current source, J_1, and I_2 is expanded to yield:

$$I_2 = V(T_{N-1}) \left[\frac{H(T_N - T_N) + 2H(T_N - T_{N-1})}{6} \right] \Delta T_N + V(T_N) \left[\frac{2H(T_N - T_N) + H(T_N - T_{N-1})}{} \right] \Delta T_N \tag{2-24}$$

Then, I_2 can be considered to be associated with a parallel combination of a variable current source, J_2, and a variable conductance, G. Letting $J = J_1 + J_2$, we can get:

$$I_{AB}(T_N) = J + V(T_N)G. \tag{2-25}$$

In terms of circuit elements this model becomes that shown in Figure 2-2.

III. CONVOY TEST CASE

TEST PLAN

In order to test the FORTRAN convolution routine, CONVOY, it was necessary to introduce a block of circuitry with a known spectral representation into a more complex network. The overall circuit itself permitted evaluation by SCEPTRE in the usual manner, and the known spectrum permitted an inverse Fourier Transform to be taken, thus supplying an h (t) for the removed block of circuitry for analysis by SCEPTRE in the convolution mode. The five major steps in the test are itemized in the following paragraphs.

Determine Circuit Representation   The test block of circuitry used was a parallel R,L,C network with the element values given by $R = 12k\Omega$, $L = 10^3\ \mu H$ and $C = 10^{-3}\ \mu F$. The values chosen gave an underdamped response characterized by the following parameters:

a. Resonance $\omega_o = (\sqrt{LC})^{-1} = 1$ MHz

b. Time Constant $\tau_o = 2\ RC = 24\ \mu s$

c. Damping Factor $\zeta = (\tau_o\ \omega_o)^{-1} = 0.04$

d. Bandwidth $B = 2(\tau_o{}^{-1}) = 80$ kHz

e. Unloaded Circuit Q $Q_o = \omega_o\ (B^{-1}) = 12$

Write Spectral Representation H (ω)   The complex frequency associated with this block was given by:

$$H(\omega) = R\ (\omega) + j\ \ I\ (\omega),$$

where

$$R\ (\omega) = \frac{G}{G^2 + B^2}\ ; \qquad I\ (\omega) = \frac{B}{G^2 + B^2}$$

and

$$G = \frac{1}{R}\ ; \quad B = (\omega C - \frac{1}{\omega L})$$

Evaluate H (ω) Versus ω   This was done using approximately 350 frequency points. The points were chosen to cover the frequency band of interest with the majority being used to define the function around resonance.

Calculate Inverse Fourier Transform Our own internally developed inverse Fourier Transform routine, FOUR2, was used to calculate the time function h(t). This routine is capable of taking non-uniform steps in both the time and the frequency domain. The time function was evaluated from 0.1 μs until 26 μs, using steps of 0.1 μs.

SCEPTRE Solution This was done with and without convolution.

TEST RESULTS

The SCEPTRE reference circuit with its convolution representation is shown in Figure 3-1. The voltages across and the currents through all the convolution terminals were compared. All of the outputs converged and had essentially died-out after approximately 1000 integration steps.

The first and second maximums of a typical output occurred after approximately 100 and 300 steps, respectively. The numerical agreement between the two runs at these two points was within 0.5 percent.

IV. SUMMARY

The convolution routine CONVOY has been written, debugged and tested within the SCEPTRE Program, and is working satisfactorily. It has been compared in both the impedance and the admittance modes with known linear and non-linear reference circuits. The multi-point interfaces were chosen to represent both driving-point and transfer configurations. The program is presently tailored to the Runge-Kutta integration routine. It can easily be extended, if so desired, to work in conjunction with the other explicit integration routines. However, incorporation into the implicit integration routine, although possible, appears to be disadvantageous.

The program is currently restricted to H(t) functions which are given in tabular form. It is also easily extended to accept input functions which are given analytically as functions of time.

The program can rapidly become core limited as the number of H(t) functions increases. It is currently limited to an approximate maximum of 50 to 70 H(t) functions which are defined, on average, by 100 or so tabular entries. This maximum is machine dependent. It also depends on the size and complexity of the remaining SCEPTRE circuitry. A possible solution to this problem would be storage of these tables on a disk. However, this method would require an increased amount of time for the input-output processing.

The effectiveness, and hence the accuracy within which the program will work, is dependent on the thoroughness of the frequency domain to time domain mapping. In our test cases the H(t) functions were accurately described, and consequently, the agreement of the results of SCEPTRE with and without CONVOY were very good. The maximum differences were only several percent after hundreds of SCEPTRE integration steps. An inadequate spectrum characterization will produce a less-than representative inverse Fourier transform, and also use of an insufficient number of points in the time domain will result in numerical calculation errors. This requirement for good time domain definition of the H functions and the previous requirement of core conservation are in direct opposition to each other. Thus, in practice, a trade-off between the two is necessary.

In conclusion, all things considered, it appears that the convolutional capability of SCEPTRE promises to be a valuable analytic tool.

ACKNOWLEDGMENT

The authors would like to express their thanks to Dr. D. S. Becker for initially suggesting this topic and for his constant encouragement throughout the work. This work was performed on Air Force Contract F04701-71-C-0080.

REFERENCES

(1) H. Mathers, S. Sedore and J. Sents, "SCEPTRE SUPPORT - VOLUME I: REVISED SCEPTRE USER'S MANUAL", International Business Machine Corporation, Technical Report No. AFWL-TR-66-124, Vol. I, April 1968.

(2) B. Hanrahan, "TRAFFIC USER'S MANUAL", Boeing Company Report Number D2-19334 Code Ident. Number 81205, November, 1970.

(3) C. Frye and M. Apfelbaum, "FOUR2 - Fourier Transformations Involving Non-Uniform Steps", (Internal Technical Note - to be published).

(4) De Russo, Roy and Close, State Variables for Engineers, John Wiley and Sons, New York, 1965.

(5) H. Mathers, S. Sedore and J. Sents, "Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation "SCEPTRE", International Business Machines Corporation, Technical Note No. AFWL-TR-66-126, Vol. II, April 1968.

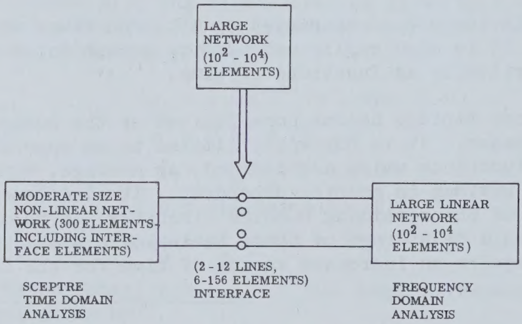

LARGE NETWORK (10^2 - 10^4) ELEMENTS)
MODERATE SIZE NON-LINEAR NET-WORK (300 ELEMENTS INCLUDING INTER-FACE ELEMENTS)
LARGE LINEAR NETWORK (10^2 - 10^4 ELEMENTS)
(2 - 12 LINES, 6-156 ELEMENTS) INTERFACE
SCEPTRE TIME DOMAIN ANALYSIS
FREQUENCY DOMAIN ANALYSIS

Figure 1-1. Separation of Linear and Non-Linear Subnetworks

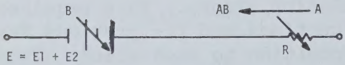

B
AB
A
E = E1 + E2
R

Figure 2-1. Impedance Model

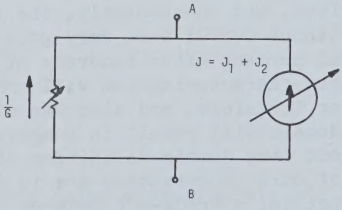

A
J = J_1 + J_2
$\frac{1}{G}$
B

Figure 2-2. Admittance Model

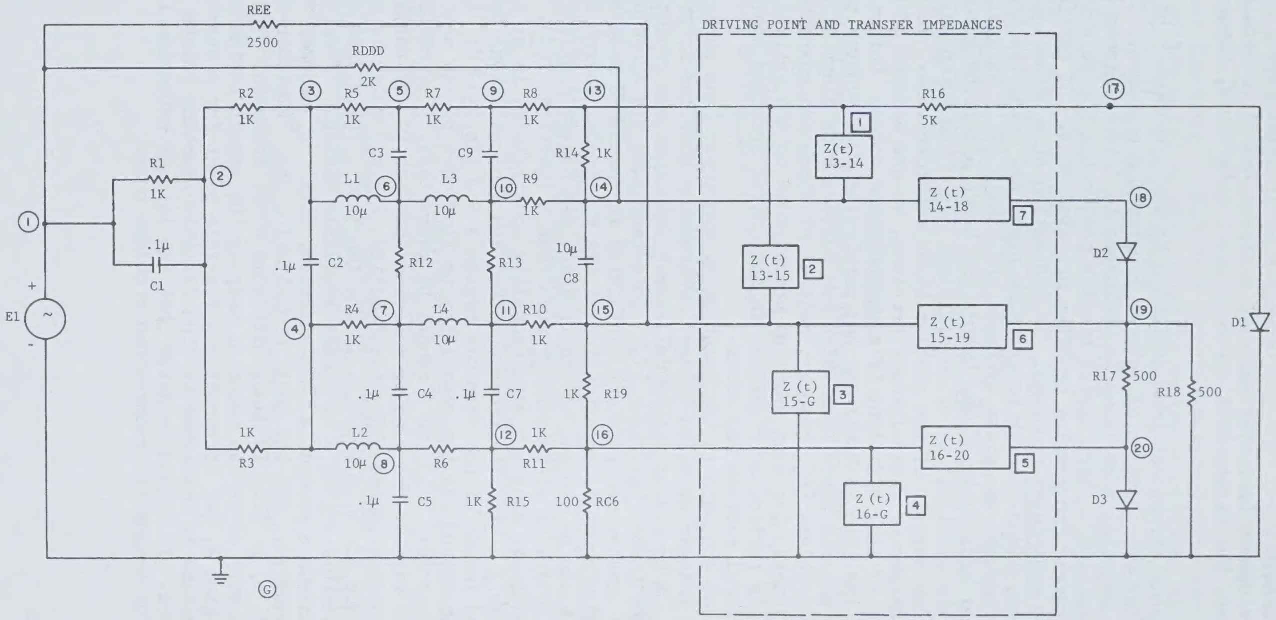

DRIVING POINT AND TRANSFER IMPEDANCES
REE
2500
RDDD
2K
R1
1K
C1
.1μ
E1
+
-
~
R2
1K
R3
1K
R4
1K
R5
1K
R6
R7
1K
R8
1K
R9
1K
R10
1K
R11
1K
R12
R13
R14
1K
R15
1K
R16
5K
R17
500
R18
500
R19
1K
RC6
100
C2
.1μ
C3
C4
.1μ
C5
.1μ
C7
.1μ
C8
10μ
C9
L1
10μ
L2
10μ
L3
10μ
L4
10μ
D1
D2
D3
Z(t)
13-14
Z (t)
13-15
Z (t)
15-G
Z (t)
16-G
Z (t)
14-18
Z (t)
15-19
Z (t)
16-20
1
2
3
4
5
6
7
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
⑭
⑮
⑯
⑰
⑱
⑲
⑳
G

Figure 3-1. SCEPTRE Reference Circuit With Convolution Representation

NONLINEAR OPTIMIZATION FOR INTEGER PROBLEMS

JASNA OPACIC
Bell Telephone Laboratories
Whippany, New Jersey 07981

DANIEL TABAK
RPI of Connecticut
Hartford, Connecticut 06120

Considerable effort has been dedicated to the development of efficient algorithms for the solution of Integer Programming problems. The vast majority of existing algorithms is dedicated to Integer Linear Programming problems [1-4]. Very little results concerning the solution of Nonlinear Integer Programming problems are available. Of particular interest are non-convex problems. An indication of a potential solution of non-convex, mixed-integer programming problems, using the branch and bound approach, was given by Beale and Tomlin [5]. Naturally the computational efficiency of any branch and bound algorithm strongly depends on the availability of an efficient algorithm which solves the continuous counterpart of the problem under consideration. In the case of Linear Integer or Mixed-Integer problems, we are assured of the availability of many efficient LP algorithms and codes. Unfortunately, this is not the case in Nonlinear and particularly non-convex problems. There exist several algorithms which can solve a wide class of nonlinear programming problems [6], however, neither of those can guarantee us the desired efficiency (or even eventual convergence to the optimal solution). The problem is particularly noticed in non-convex problems where multiplicity of local extrema may occur.

A different approach to the solution of nonlinear (and particularly non-convex) integer or mixed-integer programming problems is proposed in this paper. A heuristic multimodal search technique was recently developed and tested by Opacic [7]. This technique was subsequently extended to constrained problems, resulting in the HUMSS program [8].

The HUMSS program is an extension of the multimodal search SCAN [7], such that it can be applied to constrained, non-convex problems. The task is to determine minimizers $\underline{x}^*_k$ such that a function $f(\underline{x})$, subject to a set of nonlinear equality and inequality constraints, is at a local minimum k for $\underline{x} = \underline{x}^*_k$. The present method can find solutions both on the boundary and inside of the feasible region $R = \{\underline{x} \mid g_i(\underline{x}) \geq 0;\ i=1,\ldots,q\}$

While the method has the capability to find most of the available solutions (possibly all), this cannot be guaranteed. The HUMSS program utilizes the barrier and penalty function approach [9] by formulating a modified unconstrained objective function $P(\underline{x}, r_m)$ which is minimized for a sequence of r_m, m = 0,1,2... The function $P(\underline{x}, r_m)$ is searched over the feasible region R to locate available minima. From each obtained minimum a sequential minimization, until r_m becomes very small, is then conducted. For the local minimization the Fletcher-Powell method [ref. 6, p. 111] is used, with some novel features. The program HUMSS has a very effective means of keeping the search inside of the feasible region R, but it requires a feasible point for a start. A separate routine is developed to find feasible starting points which may be later used. It should be noted that the integer extension is performed for inequality constrained problems only.

Once a local continuous extremum has been located, its integer counterpart is found utilizing a simplex solution of a LP problem. The feasible region of this LP problem is an n-dimensional cube comprised of the integer edges of $\underline{x}^*_k$. The objective function is established in the following manner. If $\underline{x}^*_k$ is located on the boundary of any of the $g_i(\underline{x})$, the objective function will be a hyperplane parallel to the tangent hyperplane of the $g_i(\underline{x})$ at $\underline{x}^*_k$. If $\underline{x}^*_k$ is an interior point, that is $g_i(\underline{x}^*_k) > 0$ for all i, a small sphere S, of radius $\varepsilon \ll 1$ around $\underline{x}^*_k$ is considered. On the surface of S we find a point $\underline{x}_h$ with the highest value of $f(\underline{x})$, $\underline{x} \in S$. The objective function is then chosen parallel to the hyperplane tangent to S at $\underline{x} = \underline{x}_h$. The simplex solution in this case is conducted in such a manner that feasibility $\underline{x} \in R$ is checked for every basic solution encountered. As soon as an infeasible basic point $\underline{x}_B$, i+1 is found, its predecessor $\underline{x}_{Bi}$ (which is saved in the memory) becomes the sought solution. It may happen of course that the optimum solution of the LP sub-problem is also feasible with respect to R. It becomes then the local integer extremum.

References

1. A. M. Geoffrion, R. E. Marsten, Management Sc. 18, pp. 465-491, May 1972.
2. D. R. Plane, C. McMillan, "Discrete Optimization," Prentice Hall, Englewood Cliffs, N.J., 1971.
3. H. Greenberg, "Integer Programming," Academic Press, N.Y., 1971.
4. M. L. Balinski, K. Spielberg, in J.S. Aronofsky (ed.), Progress in Operations Research, Vol. III, Wiley, N.Y., 1969.
5. E.M.L.Beale, J.A. Tomlin, in J. Lawrence (ed.), Proc. of the Fifth Int. Conf. of Operations Research, pp. 447-454, Venice, 1969.
6. D. M. Himmelblau, "Applied Nonlinear Programming," McGraw Hill, N.Y., 1972.
7. J. Opacic, "A Heuristic Approach to Multimodal Function Minimization - the Global Aspect," 9th Allerton Conf., Allerton, Ill., October 1971.
8. J. Opacic, "An Approach to Constrained Optimization Problems using an Heuristic Multimodal Search," 6th Princeton Conf., Princeton, N.J., March 1972.
9. A.V. Fiacco, G.P. McCormick, "Nonlinear Programming: SUMT", Wiley, N.Y., 1968.

FURTHER RESULTS ON A LARGE CLASS OF NON-BCH CYCLIC CODES FOR MULTIPLE RANDOM-ERROR CORRECTION

HARRY T. HSU
Department of Computer Science
The Pennsylvania State University
University Park, Pa. 16802

ABSTRACT

This paper studies a large class of non-BCH cyclic codes [1] for multiple random-error correction. According to their properties, the large class of non-BCH cyclic codes are further divided into four subclasses. It is found that class I, class II, and class III codes are also suitable to be used as error-correcting codes for a compound channel. Class I codes can be easily decoded. Class II and class III codes can be easily decoded when the minimum distance is small. Class IV codes are constructed from two class II codes.

I. INTRODUCTION

This paper studies a large class of non-BCH cyclic codes for multiple random error correction. It is found that the BCH lower bound [2] alone cannot determine the minimum distance of the non-BCH cyclic codes.

The non-BCH cyclic codes are further divided into four subclasses. The class I, class II and class III cyclic codes are also suitable to be used as error correcting codes in a compound channel [3]. The reader is assumed to be familiar with the properties and the requirements of a compound channel. For those who are not familiar with the subjects, they are referred to ref. [3]. The class I and class II cyclic codes are probably the best error-correcting codes when they are used for the simultaneous correction of single burst errors of length $\leq b$ and random errors of weight ≤ 2.

The readers are assumed to be familiar with the BCH codes [4] and their decoding algorithms [4,6,7]. For those who are not familiar with the subjects, they are referred to the cited references.

Section II deals with class I codes, sections III, IV and V deal with class II, III and IV codes respectively.

Throughout this paper, the following symbols are used:

If α is a primitive root of $x^N-1=0$, then $p_i(x)$, $1 \leq i \leq (N-2)$ represents the minimum polynomial of α^i. The equation $p_i(x) = p_j(x)$ represents that α^i and α^j are conjugate roots of $p_i(x)$. Rem $(f(x)/g(x))$ represents the remainder obtained by dividing $f(x)$ by $g(x)$. Wt$(a(x))$ represents the number of non-zero terms in the polynomial $a(x)$.

II. CLASS I CYCLIC CODES

In this section, the class I codes and their decoding algorithms are discussed in details. Several examples of the class I cyclic codes are given to illustrate the construction of class I cyclic codes.

(A) The Construction of Class I Codes

(Lemma 1) The primitive element α is a root of $p_i(x^i)$.
(Proof) Since $p_i(x)$ has α^i as a root, it is obvious that α is a root of $p_i(x^i)$.

(Theorem 1) If b divides N, then $g(x) = p_b(x^b)\, p_3(x)$ generates a class I cyclic code of length N with $d_{min} \geq 5$.

(Proof) The period of $p_b(x)$ is N/b. Therefore the period of $p_b(x^b)$ $p_3(x)$ is N.

As a result of lemma 1, it can be easily seen that $p_b(x^b)$ has α, α^2, α^4 as roots. Therefore $g(x)$ has α, α^2, α^3, α^4 as roots. As a result of the BCH lower bound, the minimum distance is at least 5.

Q.E.D.

A straightforward extension of theorem 1 is stated in theorem 2.

Let $g_1(x)$ generates a non-primitive BCH cyclic code of length N/b with $d_{min} \geq 2t + 1$. Let $g_2(x) = \prod_{m=t}^{(2t-1)} p_{(2m+1)}(x)$. If α^{2m+1}, $t \leq m \leq 2t-1$ is a root of $g_1(x^b)$, then $p_{(2m+1)}(x)$ is removed as a factor of $g_2(x)$.

(Theorem 2) $g(x) = g_1(x^b)\, g_2(x)$ generates a class I cyclic code of length N with $d_{min} \geq 4t + 1$.

As a result of lemma 1, it can be easily seen that $g_1(x^b)$ has α^i, $1 \leq i \leq 2t$ as roots. Therefore $g_1(x^b)\, g_2(x)$ has α^i, $1 \leq i \leq 4t$ as roots and $d_{min} \geq 4t + 1$.

Several examples in the following illustrate the construction of class I codes. For convenience, N is assumed to be 63 and α a primitive root of $GF(2^6)$ throughout this section.

(Example 1)

If $g(x) = p_3(x^3)\, p_3(x)$, then $g(x)$ generates a cyclic code of length 63 with $d_{min} \geq 5$.

If $g(x) = p_9(x^9)\, p_3(x)$, then $g(x)$ generates a cyclic code of length 63 with $d_{min} \geq 5$.

(Example 2)

Let $g_1(x) = p_3(x)\, p_9(x)$, $g_2(x) = p_5(x)\, p_7(x)$. Then $g(x) = g_1(x^3)$ $g_2(x)$ generates a class I cyclic code of length 63 with $d_{min} \geq 9$, while $g_1(x)\, g_2(x)$ generates a BCH code of length 63 with $d_{min} \geq 5$.

(B) The Decoding Algorithm of the Class I Cyclic Codes

The class I cyclic codes are very suitable for application to a compound channel. The decoding algorithms are discussed in the following.

(B.1) The class I cyclic codes $g(x) = p_b(x^b)\, p_3(x)$ are suitable for the simultaneous correction of all the single burst errors of length $\leq b$ and random errors of weight ≤ 2. Example 3 is used to illustrate the decoding algorithm.

(Example 3)

Let $g(x) = p_3(x^3)\, p_3(x)$, $N = 63$, $d_{min} \geq 5$. The transmitted code polynomial $a(x)$ can be written as: $a(x) = \sum_{i=0}^{2} x^i a_i(x^3)$, where $p_3(x^3) \mid a_i(x^3)$, $0 \leq i \leq 2$. The received code polynomial $R(x)$ can be written as: $R(x) = \sum_{i=0}^{2} x^i R_i(x^3)$. If each $R_i(x^3)$ contains at most a single error, then the random error pattern is also correctible. A single burst error pattern of length ≤ 3 can be corrected in this fashion.

$R_i(x^3) = a_i(x^3) + E_i(x^3)$, $0 \leq i \leq 2$, where $E_i(x^3)$ represents the error polynomial contained in $R_i(x^3)$. Therefore

$$\text{Rem}\left(\frac{R_i(x^3)}{p_3(x^3)}\right) = \text{Rem}\left(\frac{E_i(x^3)}{p_3(x^3)}\right) \quad \text{and} \quad \text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right) = \text{Rem}\left(\frac{E_i(x)}{p_3(x)}\right).$$

$R_i(x)$ is to represent the 21 received code digits as:

$$R_i(x) = (r_i, r_{i+3}, r_{i+6}, r_{i+9}, \ldots, r_{i+57}, r_{i+60}),$$

where $i = 0$, 1 or 2. $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)$ is the remainder obtained by dividing $R_i(x)$ by $p_3(x)$.

(1) If there is only one $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right) \neq 0$, then $R_i(x)$ could contain up to two errors.

(2) If there is more than one $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)$ which is not zero, then each $R_i(x)$ contains at most a single error.

The decoding of the two different cases are discussed as:

(Case 1) If only one $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right) \neq 0$. Without loss of generality, let $\text{Rem}\,(R_2(x)/p_3(x)) \neq 0$. $R_2(x)$ could contain up to two random errors.

Let x_1 and x_2 represent the two error locators in $R_2(x)$. If $E_2(x) = x^{i_1} + x^{i_2} = p_3(x)\,Q(x) + r(x)$. Since $\beta = \alpha^3$ is a root of $p_3(x)$, $S_1 = \beta^{i_1} + \beta^{i_2} = x_1 + x_2 = \text{Rem}\,(R_2(x)/p_3(x))\,\big|_{x=\beta}$.

Since the received code polynomial $R(x) = a(x) + x^2(x^{3i_1} + x^{3i_2})$,

$$\text{Rem}\left(R(x)/_{p_3(x)}\right)\Big|_{x=\beta} = \beta^2\,(\beta^{3i_1} + \beta^{3i_2}) = \beta^2\,(x_1^3 + x_2^3) = \beta^2\,S_3.$$

Therefore $S_3 = \frac{1}{\beta^2}\,\text{Rem}(R(x)/p_3(x))\big|_{x=\beta}$. For the Galois field operations, the reader is referred to Chapter 2, Berlekamp [5].

Once S_1 and S_2 have been found, the decoding of $R_2(x)$ is exactly the same as the decoding of the double error correcting BCH codes. If $R_2(x)$ contains only a single error, then $S_3 = S_1^3$.

(3) If there are more than one $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)$ which is not zero, then each $R_i(x)$ contains exactly a single error if $\text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right) \neq 0$.

$S_1 = \text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)\Big|_{x=\beta}$ gives the location of the single error digit at $R_i(x)$. A single burst error of length ≤ 3 can be decoded in this fashion.

(B.2) The decoding algorithm of the class I cyclic codes with $d_{\min} \geq 7$.

Example 4 is to illustrate the decoding algorithm. It will first be shown that the cosets of burst errors are disjoint from the cosets of random errors.

(Example 4)

Let $g_1(x) = p_3(x)\,p_9(x)$, $g_2(x) = p_5(x)\,p_7(x)$, $N = 63$. The cyclic code $g(x) = g_1(x^3)\,g_2(x)$ is capable of the simultaneous correction of all the single burst errors of length ≤ 6 and random errors of weight ≤ 4.

The following shows that the cosets of burst errors and the cosets of random errors are disjoint.

Suppose that $x^i B(x) \equiv E(x) \pmod{g(x)}$, where $L(B(x)) \leq 6$, and $wt(E(x)) \leq 4$. Then $a(x) = x^i B(x) + E(x)$ must be a code polynomial and $wt(a(x)) \leq 10$. The code polynomial $a(x)$ can be written as:

$$a(x) = a_0(x^3) + x\, a_1(x^3) + x^2\, a_2(x^3) .$$

Since $g_1(x^3) \mid a_i(x^3)$, $i = 0, 1, 2$. If $a_i(x^3) \neq 0$, then $wt(a_i(x^3)) \geq 5$. Since $wt(a(x)) \leq 10$, at least one $a_i(x^3)$, $i = 0, 1, 2$ must be zero. Without loss of generality, let $a_2(x^3) = 0$, then $wt(x^i B(x)) \leq 8$. Therefore at least two $a_i(x^3)$'s must be zero. Assume that $a_1(x^3) = a_2(x^3) = 0$, then $wt(B(x)) \leq 2$, and $wt(a(x)) \leq 6$. Since $g_1(x^3)\, g_2(x) \mid x^i B(x) + E(x)$, $wt(a(x)) \geq 9$, which is a contradiction. Therefore the cosets of burst errors and the cosets of random errors are disjoint from each other.

The complete decoding system requires two decoders. Figure 1 shows a complete decoding system. If the syndrome pattern in the burst-error decoder is never recognized, then the output of the random error decoder is taken. Otherwise the output of the burst error decoder is taken.

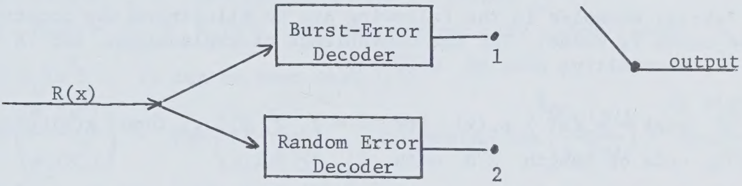


Figure 1. The Complete Decoding System

The generator polynomial used in the burst error decoder is $g_1(x^3)\, g_2(x)$. The generator polynomial used in the random error decoder is $p_1(x)\, p_3(x)\, p_5(x)\, p_7(x)$.

All the other class I codes can be decoded by the same principle as that given in examples 3 and 4.

III. CLASS II CYCLIC CODES

This section presents a class II cyclic code of the form: $g(x) = g_1(x^b)\, g_2(x)$, where g.c.d. $(b,N) = 1$, $g_1(x)\, g_2(x)$ generates a cyclic code of length N. At the moment, no good random error decoding algorithm has been found for the class II cyclic codes when $t > 4$. The decoding algorithm is reported only for the case $t = 2$. The class II codes are also suitable for application to a compound channel.

(A) The Construction of Class II Codes

(Theorem 3) If $bi \equiv 3 \pmod{N}$, then $g(x) = p_1(x^b)\, p_i(x)$ generates a class II cyclic code of length bN with $d_{min} \geq 5$.

(Proof) The transmitted code polynomial $a(x)$ can be written as:

$$a(x) = \sum_{i=0}^{(b-1)} x^i\, a_i(x^b) , \text{ where } p_1(x^b) \mid a_i(x^b) ,\ 0 \leq i \leq b - 1 .$$

If $a_i(x^b) \neq 0$, then $wt(a_i(x^b)) \geq 3$.

(1) If there are at least two $a_i(x^b)$'s which are not zeros, then $wt(a(x)) \geq 6$.

(2) If there is only one $a_i(x^b)$ which is not zero, then $a(x) = x^i a_i(x^b)$, $0 \leq i \leq (b-1)$. Since $p_1(x^b) \mid a_i(x^b)$, $p_1(x) \mid a_i(x)$. Therefore, $a_i(x)$ has $\alpha, \alpha^2, \alpha^3$ and α^4 as roots. Since $p_i(x) \mid a_i(x^b)$ and $bi \equiv 3 \pmod{N}$, $a_i(x)$ must have α^3 as root. Therefore $a_i(x)$ must have $\alpha, \alpha^2, \alpha^3$ and α^4 as roots. As a result of the BCH lower bound, $a(x)$ must have a weight ≥ 5 .

Q.E.D.

A straight-forward extension of theorem 3 is stated in theorem 4. Let $g_1(x)$ generates a BCH code of length N with $d_{min} \geq 2t + 1$. $S = \{\alpha^i \mid 2t + 1 \leq i \leq 4t - 1$ and α^i is not a root of $g_1(x)\}$. Let $g_2(x) = \Pi\, p_{j_i}(x)$ for all j_i such that $b.j_i \equiv i \pmod{N}$, where α^i is an element of S and $g_2(x)$ does not contain any repeated roots of $x^N - 1 = 0$.

(Theorem 4) If $g(x) = g_1(x^b)\, g_2(x)$, then $g(x)$ generates a class II cyclic code of length bN with $d_{min} \geq 4t + 1$.

Several examples in the following are to illustrate the construction of the class II codes. For the convenience of explanation, let $N = 31$ and α , a primitive root of $GF(2^5)$.

(Example 5)

If $g(x) = p_1(x^{2^\ell})\, p_{3^\ell}(x)$ for $\ell = 1, 2, 3, \ldots$, then $g(x)$ generates a cyclic code of length $2^\ell N$ with $d_{min} \geq 5$.

If $5\, i \equiv 3 \pmod{31}$, then $i \equiv 13 \pmod{31}$. Since $p_{13}(x) = p_{11}(x)$, $g(x) = p_1(x^5)\, p_{11}(x)$ generates a cyclic code of length 155 with $d_{min} \geq 5$.

If $6\, i \equiv 3 \pmod{31}$, then $i \equiv 16 \pmod{31}$. Since $p_{16}(x) = p_1(x)$, $g(x) = p_1(x^6)\, p_1(x)$ generates a cyclic code of length 186 with $d_{min} \geq 5$.

(Example 6)

Let $g_1(x) = p_1(x)\, p_3(x)$. If $5\, i \equiv 5 \pmod{31}$, then $i \equiv 1 \pmod{31}$. If $5\, i \equiv 7 \pmod{31}$, then $i \equiv 20 \pmod{31}$. Since $p_5(x) = p_{20}(x)$, $g_2(x) = p_1(x)\, p_5(x)$. Then $g(x) = g_1(x^5)\, g_2(x)$ generates a cyclic code of length 155 with $d_{min} \geq 9$.

(Example 7)

Let $g_1(x) = p_1(x)\, p_3(x)\, p_5(x)$. If $9\, i \equiv 7 \pmod{31}$, then $i = 18$. If $9\, i \equiv 9 \pmod{31}$, then $i = 1$. Since α^{18} is a root of $p_5(x)$, $g_2(x)$ is chosen to be $g_2(x) = p_1(x)\, p_5(x)$. Then $g(x) = g_1(x^9)\, g_2(x) = p_1(x^9)\, p_3(x^9)\, p_5(x^9)\, p_1(x)\, p_5(x)$ generates a cyclic code of length 279 with $d_{min} \geq 11$.

The proof of example 7 is very simple. Every transmitted code polynomial $a(x)$ can be written as: $a(x) = \sum_{i=0}^{8} x^i a_i(x^9)$. Each $a_i(x)$ has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^8, \alpha^{10}$ as roots.

(a) If there are at least two $a_i(x^9)$'s which are not zeros, then $d_{min} \geq 14$.

(b) If there is only one $a_i(x^9)$ which is not zero, then $p_1(x)\, p_5(x) \mid a_i(x^9)$. Then $a_i(x)$ also has α^7 and α^9 as roots. Therefore $a_i(x)$ has α^i , $1 \leq i \leq 10$ as roots. As a result of the BCH lower bound, $d_{min} \geq 11$.

(B) The Decoding Algorithm of Class II Codes

The class II codes are also suitable for application to a compound channel. However, there is no good random error decoding algorithm when $t > 4$. The random error decoding algorithm is to be explained only for the case $t = 2$. Example 8 is to illustrate the decoding algorithm.

(Example 8)

Let $p_1(x) = x^3 + x + 1$, $N = 7$. Then $g(x) = p_1(x^5)\, p_1(x)$ generates a class II cyclic code of length 35 with $d_{min} \geq 5$. The class II code is capable of the simultaneous correction of all the signle burst errors of length ≤ 5 and random errors of weight ≤ 2 .

Let $R(x) = \sum_{i=0}^{4} x^i R_i(x^5)$ be the received code vector, where $R_i(x^5) = a_i(x^5) + E_i(x^5)$. It can be seen that

$$\text{Rem}\left(\frac{R_i(x^5)}{p_1(x^5)}\right) = \text{Rem}\left(\frac{E_i(x^5)}{p_1(x^5)}\right) . \quad \text{Therefore} \quad \text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right) = \text{Rem}\left(\frac{E_i(x)}{p_1(x)}\right).$$

(Case 1) If only one $\text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right) \neq 0$, then up to three random errors can be corrected.

Let $E_i(x) = x^{i_1} + x^{i_2} + x^{i_3}$, then

$$S_1 = \text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right)\Big|_{x=\alpha} = \alpha^{i_1} + \alpha^{i_2} + \alpha^{i_3} .$$

Since $\text{Rem}\left(\frac{R(x)}{p_1(x)}\right) = x^i \ \text{Rem}\left(\frac{x^{5i_1} + x^{5i_2} + x^{5i_3}}{p_1(x)}\right)$

$$S_5 = \alpha^{5i_1} + \alpha^{5i_2} + \alpha^{5i_3} = \frac{1}{\alpha^i} \ \text{Rem}\left(\frac{R(x)}{p_1(x)}\right)\Big|_{x=\alpha} .$$

Since $\alpha^{10} = \alpha^3$, $S_3 = S_5^{\,2}$. Once S_1 , S_3 and S_5 have been found, the decoding is exactly the same as the triple error-correcting BCH codes.

(Case 2) If more than one $\text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right)$ is not zero. Each $R_i(x)$ where $\text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right) \neq 0$ contains a single error. S_1 gives the location of the error digit in $R_i(x)$.

A single burst error of length ≤ 5 can be corrected in this fashion. If each $R_i(x^5)$, $0 \leq i \leq 4$ contains at most a single error, then up to five random errors can be corrected in this fashion.

For class II codes with $d_{min} \geq 7$, by using the same approach as that given in example 4, it can be easily shown that the cosets of burst errors and the cosets of random errors are disjoint. As a result, the class II codes with $d_{min} \geq 4t + 1$ which is generated by $g_1(x^b)\ g_2(x)$ is capable of the simultaneous correction of all the single burst errors of length $\leq tb$ and random errors of weight $\leq 2t$. The class II codes are therefore suitable for application to a compound.

IV. CLASS III CYCLIC CODES

A class III code is very much like a class II code. Both have a generator polynomial of the form: $g(x) = g_1(x^b)\ g_2(x)$. For a class II code, $g_1(x)$ generates a BCH code of length N with $d_{min} \geq 2t + 1$. For a class III code, $g_1(x)$ generates a class I cyclic code of length N with $d_{min} \geq 2t + 1$. A class III code is also suitable for application to a compound channel. However, the random error decoding algorithm becomes very complicated when $t > 4$.

Several examples in the following should make the construction of class III codes clear. For convenience, N is assumed to be 63 and α, a primitive root of $GF(2^6)$.

(Example 9)

The polynomial $g_1(x) = p_3(x^3)\ p_3(x)$ generates a class I cyclic code of length 63 with $d_{min} \geq 5$. Let $g_2(x) = p_5(x)\ p_7(x)$. Then $g(x) = g_1(x^4)\ g_2(x) = p_3(x^{12})\ p_3(x^4)\ p_5(x)\ p_7(x)$ generates a class III cyclic code of length 252 with $d_{min} \geq 9$.

For the proof, the reader is referred to section III.

Since $g_1(x)$ is capable of correcting a single burst error of length ≤ 3, the class III code is capable of correcting all the single burst errors of length ≤ 12.

By using the same approach as that given in example 4, it can be easily shown that the cosets of all the single burst errors of length ≤ 12 and the cosets of all the random errors of weight ≤ 3 are disjoint from each other. Therefore the class III code is capable of the simultaneous correction of all the single burst errors of length ≤ 12 and random errors of weight ≤ 3.

(Example 10)

Let $g_1(x) = p_3(x^3)\ p_3(x)$. If $5i \equiv 5 \pmod{63}$, then $i \equiv 1 \pmod{63}$. If $5i \equiv 7 \pmod{63}$, then $i \equiv 14 \pmod{63}$. Since $p_{14}(x) = p_7(x)$, choose $g_2(x) = p_1(x)\ p_7(x)$. Then $g(x) = g_1(x^5)\ g_2(x) = p_3(x^{15})\ p_3(x^5)\ p_1(x)\ p_7(x)$ generates a class III cyclic code of length 315 with $d_{min} \geq 9$.

The class III code is capable of the simultaneous correction of all the single burst errors of length ≤ 15 and random errors of weight ≤ 3.

(Example 11)

Let $g_1(x) = p_9(x^9)\ p_3(x)$, $g_2(x) = p_1(x)\ p_7(x)$. Then $g(x) = g_1(x^5)\ g_2(x) = p_9(x^{45})\ p_3(x^5)\ p_1(x)\ p_7(x)$ generates a class III cyclic code of length 315 with $d_{min} \geq 9$.

The cyclic code is capable of the simultaneous correction of all the single burst errors of length ≤ 45 and random errors of weight ≤ 3.

Example 12 in the following is to illustrate the random error decoding algorithm for the class III codes with $t = 3$.

(Example 12)

Let $g_1(x) = p_9(x^9)\, p_3(x)$, $g_2(x) = p_1(x)$. Then $g(x) = g_1(x^5)\, g_2(x) = p_9(x^{45})\, p_3(x^5)\, p_1(x)$ generates a class III cyclic code of length 315 with $d_{min} \geq 7$. It can be easily seen that $p_1(x)\, p_3(x) \mid g_1(x)$.

The received code polynomial $R(x)$ can be written as:

$R(x) = \sum_{i=0}^{4} x^i R_i(x^5)$, where $R_i(x^5) = a_i(x^5) + E_i(x^5)$, $0 \leq i \leq 4$.

Then $\text{Rem}\left(\frac{R_i(x^5)}{g_1(x^5)}\right) = \text{Rem}\left(\frac{E_i(x^5)}{g_1(x^5)}\right)$. Therefore

$\text{Rem}\left(\frac{R_i(x)}{g_1(x)}\right) = \text{Rem}\left(\frac{E_i(x)}{g_1(x)}\right)$.

(1) If there is only one $\text{Rem}\left(\frac{R_i(x)}{g_1(x)}\right)$ which is not zero, then $R_i(x)$ could contain up to three random errors.

Let $E(x) = x^i(x^{5i_1} + x^{5i_2} + x^{5i_3}) = x^i E_i(x^5)$. Then $E_i(x) = x^{i_1} + x^{i_2} + x^{i_3}$. It can be easily seen that $S_1 = \text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right)\Big|_{x=\alpha}$, and $S_3 = \text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)\Big|_{x=\alpha^3}$. Since $\text{Rem}\left(\frac{R(x)}{p_1(x)}\right)\Big|_{x=\alpha} = \alpha^i\,(\alpha^{5i_1} + \alpha^{5i_2} + \alpha^{5i_3}) = \alpha^i S_5$, $S_5 = \frac{1}{\alpha^i}\, \text{Rem}\left(\frac{R(x)}{p_1(x)}\right)\Big|_{x=\alpha}$.

Once S_1 , S_3 and S_5 have been found, the decoding procedure is exactly the same as the triple-error-correcting BCH codes.

(2) If there are at least two $\text{Rem}\left(\frac{R_i(x)}{g_1(x)}\right)$'s which are not zeros.

Then each $R_i(x)$ where $\text{Rem}\left(\frac{R_i(x)}{g_1(x)}\right) \neq 0$ can contain at most two random errors. $S_1 = \text{Rem}\left(\frac{R_i(x)}{p_1(x)}\right)\Big|_{x=\alpha}$ and $S_3 = \text{Rem}\left(\frac{R_i(x)}{p_3(x)}\right)\Big|_{x=\alpha^3}$. Once S_1 and S_3 have been found, the decoding procedure is exactly the same as the double error correcting BCH code.

V. CLASS IV CYCLIC CODES

For every class II code of length bN with $d_{min} \geq 4t + 1$, there always exists a class IV cyclic code of length $2bN$ with $d_{min} \geq 4t + 1$. The class IV code constructed has a better rate of transmission than the class II code.

(Lemma 2) If $\deg(a(x)) \geq N$ and $a(x) \equiv a_1(x) \pmod{x^N + 1}$, where $\deg(a_1(x)) < N$, then $wt(a_1(x)) \leq wt(a(x))$.

Lemma 2 is an obvious result, no proof is therefore supplied.

(Lemma 3) If $g_1(x^b)\, g_2(x) \mid x^{bN} + 1$ and $g_{11}(x^b)\, g_{12}(x) \mid g_1(x^b)$, then

$$\text{g.c.d.}\left(\frac{x^{bN}+1}{g_1(x^b)\,g_2(x)}\ ,\ g_{11}(x^b)\,g_{12}(x)\right) = 1\ .$$

(Proof) Suppose not, then $x^N + 1 = 0$ contains a repeated root. Since $N \mid 2^m - 1$, clearly it is a contradiction.

Q.E.D.

(Theorem 4) If $f_1(x) = g_1(x^b)\,g_2(x)$ generates a class II cyclic code of

* (Continued on last page)

(Proof) Suppose that $d_{min} \leq 4t$, then there exists a non-zero code polynomial $a(x)$ of length $2bN$ with $wt(a(x)) \leq 4t$. $a(x)$ can be written as: $a(x) = (x^{bN} + 1)\,Q(x) + a_1(x)$, where $\deg(a_1(x)) \leq bN-1$. As a result of lemma 2, $wt(a_1(x)) \leq wt(a(x)) \leq 4t$. Since $f_1(x) \mid a(x)$ and $f_1(x) \mid x^{bN} + 1$, $f_1(x) \mid a_1(x)$. If $a_1(x) \neq 0$, then $wt(a_1(x)) \geq 4t + 1$, which is clearly a contradiction. Therefore $a_1(x) = 0$ and $a(x) = (x^{bN} + 1)\ Q(x)$, where $\deg(Q(x)) \leq bN-1$. As a result of lemma 3, $f_2(x) \mid Q(x)$. Therefore $wt(Q(x)) \geq 2t + 1$. Then $wt(a(x)) \geq 4t + 2$, which is a contradiction. Therefore all the non-zero code polynomials are of weight $\geq 4t + 1$.

Q.E.D.

The following gives an example of the class IV cyclic codes.

(Example 13)

If $N = 31$, then $f_1(x) = p_1(x^5)\,p_3(x^5)\,p_1(x)\,p_5(x)$ generates a class II cyclic code of length 155 with $d_{min} \geq 9$. The polynomial $f_2(x) = p_1(x^5)\,p_{11}(x)$ generates a class II cyclic code of length 155 with $d_{min} \geq 9$. Since $5 \times 11 \equiv 24 \pmod{31}$ and $p_3(x) = p_{24}(x)$, $f_2(x) \mid p_1(x^5)\,p_3(x^5)$. Therefore $g(x) = f_1(x)\,f_2(x)$ generates a class IV cyclic code of length 310 with $d_{min} \geq 9$.

At the moment, no decoding algorithm for class IV codes has been found yet.

REFERENCES

[1] Hsu, H. T., "A Large Class of Cyclic and Shortened Cyclic Binary Codes for Multiple Error Correction," Information and Control, Vol. 16, No. 3, pp. 231-242, 1970.

[2] Peterson, W. W., ERROR-CORRECTING CODES, The M.I.T. Press, 1961.

[3] Hsu, H. T., T. Kasami, and R. T. Chien, "Error-Correcting Codes for a Compound Channel," IEEE Trans. on Inf. Theory, IT-14, pp. 135-139, Jan. 1968.

[4] Bose, R. C., and D. K. Ray-Chauhuri, "On a Class of Error Correcting Binary Group Codes," Inf. and Control, 3, 68-79 (1960).

[5] Berlekamp, E. R., Algebraic Coding Theory, McGraw-Hill, New York, 1968.

[6] Chien, R. T., "Cyclic Decoding Procedure for the Bose-Chauhuri-Hocquenghem Codes," IEEE Trans. on Inf. Theory, IT-10, pp. 357-363, Oct. 1964.

[7] Massey, J. L., "Shift-Register Synthesis and BCH Decoding," IEEE Trans. on Inf. Theory, IT-5, pp. 122-127, Jan. 1969.

---

* length bN with $d_{min} \geq 4t + 1$, $f_2(x) = g_{11}(x^b)\, g_{12}(x)$ generates a class II cyclic code of length bN with $d_{min} \geq 2t + 1$ and $f_2(x) \mid g_1(x^b)$, then $g(x) = f_1(x)\, f_2(x)$ generates a class IV code of length 2bN with $d_{min} \geq 4t + 1$.

A NOTE ON SYSTEMATIC NON-SEPARATE ARITHMETIC CODES

M. R. VARANASI AND T. R. N. RAO
Electrical Engineering Department
University of Maryland
College Park, Maryland 20742

ABSTRACT

This paper extends the results of a previously published paper on systematic subcodes. In that paper, an approach for the construction of systematic nonseparate codes, namely $|gAN|_M$ codes, from the non-systematic AN codes is given. Also, if the systematic distance defined on the $|gAN|_M$ codes is a metric, then the error control properties of AN codes are preserved in the systematic $|gAN|_M$ codes. We consider here the case when the systematic distance defined on the $|gAN|_M$ code $(M=A\cdot m=A\cdot 2^k)$ is not a metric. Then its error control properties are exactly the same as the AN code if and only if

$$m < M(A, 2t+1) \quad \text{for} \quad t = 1$$

and

$$m < 1/2\ M(A, 2t+1) \quad \text{for} \quad t > 1.$$

INTRODUCTION AND BACKGROUND

An arithmetic code is a selected set of code words or code numbers each of, say, n digits, (in some radix r representation). For each integer N from the information set $Z_m = \{0, 1, \ldots, m-1\}$, there is a corresponding codeword. Errors can be detected or corrected in arithmetic operations using these code words [3]. An arithmetic code is systematic if and only if some specified k digits $(k < n)$ of the code words are the information and the remaining n-k digits represent one or more checks. Systematic codes can be divided into separate and nonseparate categories. In codes that are non-systematic the information and check digits are unidentifiable without some form of decoding operation.

Based on the preceding, arithmetic codes fall into three major classes: 1) AN codes ([3],[4],[5]) which are non-systematic and therefore non-separate; 2) separate codes ([6],[7],[9]) with one or more residue checks, for example the $(N, |N|_A, |N|_B)$ biresidue codes and 3) the systematic nonseparate codes ([10],[14], [15]) which are also called systematic subcodes.

Garner [10] has derived the conditions for the existence of a class of systematic nonseparate codes that are derivable from the non-systematic AN codes. Rao ([8],[1]) has introduced "systematic weight" on the systematic non-separate codes and established the error control properties of the systematic subcodes to the minimum systematic weight of the code whenever the systematic distance satisfies the properties of a metric. In this paper we try to extend the above results when the systematic distance is not a metric. A brief review of the above results is presented for the sake of completeness. To maintain consistency in terminology and notation with [1], we define the symbols as follows. $M(A,d)$: the least positive integer such that the arithmetic weight of the product $A \cdot M(A,d)$ is less than d,

m: the information range, $0 \leq N < m$

M: code range $M = A \cdot m$

$Z_M = \{0, 1, \ldots M-1\}$

Systematic weight $\xi(N)$ of $N \in Z_m$ is the minimum number of terms in the representation of N or $\overline{N} = M-N$ of the form

$$N \text{ or } \overline{N} = a_1 V_{j_1} + a_2 V_{j_2} + \ldots .$$

where $a_i = 1$ or -1 and V_{j_i} are the digit values as defined in [10]. When $m = 2^k$, $V_j = 2^j$ for $j = 0, 1, \ldots n-1$, and $\xi(N) = \xi(\overline{N}) = \min\{AW(N), AW(\overline{N})\}$ and the systematic weight then is the same as the modular arithmetic weight (MAW) previously defined by Rao and Garcia [9].

The systematic distance between $x, y \in Z_M$ denoted as $D(x,y)$ is given by $\xi(x-y)$.

Furthermore, throughout this paper we use m of the form 2^k so that $M = A \cdot m = A \cdot 2^k$ for some k.

Let us examine the construction of systematic subcodes from the AN codes. Garner [10] gives a procedure as follows. Consider an AN code for $0 \leq N < m$ such that A and m are relatively prime. Then there exists a $|gAN|_M$ code where

$$M = A \cdot m \tag{1}$$

and

$$g \cdot A = C_1 m + 1 \ , \quad 0 < C_1 < A \quad . \tag{2}$$

The codeword representing 1, gA, can also be written as a pair $(C_1, 1)$. From (2) we get $|C_1 m+1|_A = 0$ or

$$C_1 = |-m^{-1}|_A \tag{3}$$

Consequently, the codeword representing N(for $0 \le N < m$) is $|gAN|_M = |C_1 \cdot N \cdot m + N|_M = |C_1 N|_A \cdot m + N$. Denoting

$$C_N = |C_1 N|_A = |-N \cdot m^{-1}|_A \ , \tag{4}$$

we have

$$|gAN|_M = C_N \cdot m + N \tag{5}$$

which can be represented as a pair (C_N, N).

Let $x = |gAN|_M = C_N \cdot m + N$ be a codeword of N .

Then

$$x = \sum_{j=0}^{n-1} n_j V_j \tag{6}$$

where $V_j = 2^j$, $j = 0, 1, \ldots k-1$

$$V_j = 2^{j-k} \quad , \ j = k \ , \ k+1, \ldots, n-1 \tag{7}$$

When $m = 2^k$, the digit values become conventional digit values of the pure binary system and $V_j = 2^j$ for all $j = 0, 1, \ldots n-1$. The complementation of a codeword (C_N, N) can be obtained as $(C_{\overline{N}}, \overline{N})$. The systematic distance of the $|gAN|_M$ code is given by the following theorem [1].

Theorem 1: A $|gAN|_M$ code has minimum systematic distance d, provided

$$m < M(A, d) \ . \tag{8}$$

The systematic distance is a metric function if it satisfies the following three properties for all $x, y \in Z_M$. (See, for instance, Birkhoff and McLane[12])

$$D(x, y) = D(y, x) \qquad \text{(symmetry)} \tag{9}$$

$$D(x, y) \ge 0 \ , \ \text{equality holds iff } x = y \ \text{(positive definite)} \tag{10}$$

$$D(x, y) + D(y, w) \ge D(x, w) \qquad \text{(triangle inequality)} \tag{11}$$

SYSTEMATIC DISTANCE AND ERROR CONTROL PROPERTIES

When the systematic distance is a metric function, the error control properties of the $|gAN|_M$ code are related to it as follows [1].

Theorem 2: Assume the systematic distance satisfies the triangle inequality in Z_M. Then the $|gAN|_M$ code can detect any error pattern of weight d or less ($\xi(E) \leq d$) iff the minimum systematic distance is at least d+1. Also this code can correct any error of weight t or less ($\xi(E) \leq t$) iff the minimum systematic distance is at least 2t+1.

It is further stated that if A and M are such that additions modulo A and modulo m each require two or fewer EAC's, then the systematic distance satisfies (11) in Z_M. On the other hand, when the systematic distance does not satisfy (11) in Z_M, the following results from [2] state the relationship between the minimum systematic distance of the code and the error control properties.

Lemma 1: A $|gAN|_M$ code in Z_M can detect any error pattern of weight d or less ($\xi(E) \leq d$) iff the minimum systematic distance of the code is at least d+1.

Proof: Let E be any error pattern of weight d or less. $|E|_A$ is nonzero since $|E|_A = 0$ implies $\xi(E) \geq d+1$. Conversely, assume $|E|_A \neq 0$ for any $E \varepsilon Z_M$. Therefore E is not a multiple of A and hence it is not a codeword. Also, the union of error sets of weight less than d exhausts all the nonzero integers in Z_M having systematic weight less than d+1. Thus every nonzero codeword must have a weight at least d+1.

Theorem 3: Given that a $|gAN|_M$ code satisfies

$$m < 1/2\ M(A, 2t+1) \qquad (12)$$

the code can correct any error pattern of weight t or less ($\xi(E) \leq t$) iff the minimum systematic distance of the code is at least 2t+1.

An outline of the proof is as follows:

Assume that the minimum systematic distance of the code is 2t+1. Consider any two distinct errors E_1 and $E_2 \varepsilon Z_M$ of systematic weight t or less ($\xi(E_i) \leq t$, i = 1, 2). If $|E_1|_A = |E_2|_A$ it follows $||E_1 - E_2|_M|_A = 0$ and so $|E_1 - E_2|_M$ must have a weight at least 2t+1.

Let E_1' and $E_2' \varepsilon Z_M$ denote respectively E_1 or $\overline{E}_1 = M - E_1$ and E_2 or $\overline{E}_2 = M - E_2$ whichever has the smaller systematic weight. Then $|E_1 - E_2|_M$ may be either $|E_1' - E_2'|_M$ or $|E_1' + E_2'|_M$.

Since the expansions of E_1' and E_2' have at most t terms in their representation, $|E_1'-E_2'|_M$ and $E_1'+E_2'$ will have at most 2t terms in their representation. Also, $E_1'+E_2' < 2M$. Any multiple of A less than $A \cdot M(A, 2t+1)$ and hence less than 2M is a codeword and therefore has at least 2t+1 terms in its representation. Since $A|M$, A does not divide $|E_1'-E_2'|_M$ or $|E_1'+E_2'|_M$ and therefore does not divide $|E_1-E_2|_M$ and consequently $|E_1-E_2|_A \neq 0$. Thus $|E_1|_A \neq |E_2|_A$ and therefore their syndromes are distinct and they are correctable.

Conversely assume that all errors of weight t or less are correctable. If there exists a codeword X of systematic weight less than or equal to 2t, by appropriate separation of the terms of X, it is expressable as two terms E_1 and E_2 such that $\xi(E_i) \leq t$ for i = 1, 2. Therefore $|E_1|_A$ and $|\bar{E}_2|A$ are not distinct and hence are uncorrectable. This is contrary to our assumption. Therefore every non-zero codeword should be of weight at least 2t+1.

We now proceed to show that the constraint on the information range given by (12) is not necessary for single error correction but it is necessary for correction of errors of weight $t > 1$.

Theorem 4: A $|gAN|_M$ code with minimum systematic distance of 3 for its nonzero codewords can correct any single error pattern $(\xi(E) = 1)$ provided

$$m < M(A, 3) \tag{13}$$

Proof: Let E_1 and $E_2 \epsilon Z_M$ be two error patterns such that $\xi(E_1) = \xi(E_2) = 1$ and let us assume without loss of generality that $E_1 > E_2$.

Now, $\quad \xi(E_1) = 1 \Rightarrow E_1 = 2^j$ or $M-2^j$ (14)

and $\quad \xi(E_2) = 1 \Rightarrow E_2 = 2^\ell$ or $M-2^\ell$. (15)

Assume that $|E_1|_A = |E_2|_A$ which implies $|E_1-E_2|_A=0$ and therefore $|E_1-E_2|_M$ is a codeword. Since A divides M, $|E_1-E_2|_A=0$ can be either of the two forms $|2^j+2^\ell|_A=0$ or $|2^j-2^\ell|_A=0$. It is impossible for the latter case to be zero. The former case will hold only if $2^j+2^\ell = 2^e \pm 1$*

---

*e is the smallest integer such that $2^e \equiv \pm 1 (\text{mod } A)$ and $AM(A, 3) = 2^e \pm 1$.

which is again impossible because of the limits on E_1 and E_2. Consequently $|E_1 - E_2|_A \neq 0$ and therefore $|E_1|_A \neq |E_2|_A$. Thus E_1 and E_2 have distinct syndromes and so they are correctable.

However for t-fold error correction ($\xi(E) \leq t$, $t > 1$) the constraint on the information range, given by inequality (12), is necessary as shown by the following theorem.

Theorem 5: Given $A \cdot M(A, 2t+1) = 2^e \pm 1$ for some $t > 1$, the $|gAN|_M$ code can correct all error patterns of weight t or less iff

$$m < 1/2\ M(A, 2t+1) \tag{16}$$

Proof: Theorem 3 establishes the fact that the inequality (16) is sufficient for t-fold error correction in $|gAN|_M$ codes. We only prove here the necessity for (16).

Let the $|gAN|_M$ code be capable of correcting all error patterns of weight t or less. Assume for the above code (16) does not hold. That is, $M(A, 2t+1) > m > 1/2\ M(A, 2t+1)$. We proceed to show that there exist two distinct errors E_1 and $E_2 \varepsilon Z_M$ each of systematic weight t or less such that $|E_1 \pm E_2|_M$ is a codeword. This leads us to show that $|E_1|_A = |\overline{E}_2|_A$ (where $\overline{E}_2 = M - E_2$) and therefore E_1 and E_2 do not have distinct syndromes and hence they are not correctable which contradicts our assumption.

Let $E_1 = 2^{e-1}$ and $E_2 = 2^{e-1} \pm 1$ such that $E_1 \neq E_2$, $E_1 \neq M - E_2$ and $\xi(E_i) \leq t$ for $i = 1, 2$. It follows from the above that $E_1 + E_2 = 2^e \pm 1 = A \cdot M(A, 2t+1)$.

Therefore $|E_1 + E_2|_M = 2^e \pm 1 - M = AN$ for some $N < M(A, 2t+1)$. Since AN is a multiple of A and less than $A \cdot M(A, 2t+1)$ it is a codeword and has at least $2t+1$ terms in its representation. This, of course, is possible because of a number of end-around-carries in addition modulo M. On the other hand if the end-around-carries are 2 or less, the systematic distance is a metric and this would not have been possible. Thus we have

$$|E_1 \pm E_2|_M = AN \quad \text{so that} \quad |E_1 \pm E_2|_A = 0\ , \tag{17}$$

and therefore

$$|E_1|_A = |\overline{E}_2|_A$$

Thus E_1 and E_2 are two distinct errors of weight t or less such that their syndromes $|E_1|_A$ and $|\bar{E}_2|_A$ are not distinct and therefore they are not correctable. This is contradictory to our assumption. Therefore (16) must hold.

Example: Let $B = 19$ and $A = \frac{2^{e(B)}-1}{B}$. Then we have $e(B) = 18$ and $A = \frac{2^{18}-1}{19} = 13{,}797$. The minimum distance of the AN code for which A is the generator and $0 \leq N < 19$ as the information range is given by the integer portion of $\left[\frac{19+1}{3}\right] = 6$. The AN code is therefore capable of double error correction and $M(A, 6)$ for this code is equal to 19 (see [13]). Let us choose $m = 2^k$ and $M(A, 6) > m > 1/2\, M(A, 6)$ so that $m = 2^4 = 16$. The corresponding $|gAN|_M$ code is obtained as $|13\, AN|_M$ code. The single error $E_1 = 2^{17}$ and the double error $E_2 = 2^{17}-1 \in Z_M$. Note $E_1 \neq E_2$ and $E_1 \neq \bar{E}_2$. Further $|E_1+E_2|_M = |2^{18}-1|_M = 2^{18}-1-M$ which is clearly a multiple of A and is less than $A \cdot M(A, 6)$. Therefore $|E_1+E_2|_M$ is a codeword. Consequently $|E_1|_A = |\bar{E}_2|_A$. We can see that E_1 and $\bar{E}_2$ are two distinct errors in Z_M of systematic weight t or less such that their syndromes are not distinct and therefore they are not correctable.

CONCLUSION

We have considered in this paper the error control properties of systematic nonseparate codes, namely the $|gAN|_M$ codes, derived from AN codes. The error control properties of the AN code can be transferred to a $|gAN|_M$ code provided the systematic distance defined on the $|gAN|_M$ code is a metric and the information range $m < M(A, 2t+1)$. However when the systematic distance is not a metric single error correcting capability can still be transferred to the corresponding $|gAN|_M$ codes provided $m < M(A, 3)$ but error correcting capability for $t > 1$ requires that the information range m to be restricted to $1/2M(A, 2t+1)$.

REFERENCES

[1] T. R. N. Rao, "Error Correction in Adders Using Systematic Subcodes," Vol. C-21, No. 3, pp. 254-259, March 1972.

[2] A. K. Trehan, "Residue Codes and Their Application to Arithmetic Processors," Ph. D. Dissertation, Dept. of Electrical Engineering, University of Maryland, 1971.

[3] D. T. Brown, "Error Detecting and Correcting Binary Codes for Arithmetic Operations," IRE Trans. Electron. Comput., Vol. EC-9, pp. 333-337, Sept. 1960.

[4] J. M. Diamond, "Checking Codes for Digital Computers," Proc. IRE (Corresp), Vol. 43, pp. 487-488, April 1955.

[5] W. W. Peterson, "Error Correcting Codes," Cambridge, Massachusetts, MIT Press, 1961.

[6] T. R. N. Rao, "Error Checking Logic for Arithmetic Type Operations of a Processor,"IEEE Trans. Comput. Vol. C-17, pp. 845-849, Sept. 1968.

[7] T. R. N. Rao, "Biresidue Error Correcting Codes for Computer Arithmetic," IEEE Trans. Comput, Vol. C-19, pp. 398-402, May 1970.

[8] T. R. N. Rao, "On Systematic Nonseparate Arithmetic Codes," in Proc. 4th. Annual Princeton Conference on Infor. Syst. Info. Sci., March 1970.

[9] T. R. N. Rao and O. N. Garcia, "Cylcic and Multi-residue Codes for Arithmetic Operations," IEEE Trans. Inform. Theory Vol. IT-17, pp. 55-91, January 1971.

[10] H. L. Garner, "Error Codes for Arithmetic Operations," IEEE Trans. Electron. Comput. Vol. EC-15, pp. 763-770, Oct. 1966.

[11] O. N. Garcia, "Error Codes for Arithmetic and Logical Operations," Ph. D. Dissertation, Dept. of Elect. Eng., University of Maryland, College Park, Maryland, 1969.

[12] G. Birkhoff and S. McLane; A Survey of Modern Algebra, New York, Macmillan, 1953.

[13] R. T. Chien, S. J. Hong, and F. P. Preparata, "Some Results in the Theory of Arithmetic Codes," Information and Control 19, 246-264, (1971).

[14] D. S. Henderson, "Residue Class Error Checking Codes," in Proc. 16th Natl. Meeting ASS. Comput. Mach., Los Angeles, California 1961.

[15] J. L. Massey, "Survey of Residue Coding for Arithmetic Errors," Int. Computation Cen. Bull., Vol. 3, Rome, Italy, UNESCO, Oct. 1964.

CHANNEL MODELING AND BURST CHARACTERISTICS

STEPHEN TSAI
University of Illinois at Chicago Circle
Chicago, Illinois 60680

ABSTRACT

A simple partitioned Markov chain model is used to represent the real channel and the error burst is uniquely defined. The probability of a burst with n errors and of length m can be calculated from the model. This probability $P_n(m)$ is shown to be a probability density function of the two-dimensional random variable (n,m) and can facilitate the evaluation of the effectiveness of error control techniques.

I. INTRODUCTION

On a real channel, the errors do not occur independently. The performance of the error control technique depends on how the errors occur. A tremendous amount of error data has been collected on a variety of channels, e.g. HF channel and troposcatter channel. Much effort has been made in finding a mathematical model from which to derive the error statistic similar to that of a real channel. Gilbert[1], Berkovits and Cohn[2], [3], [4], Berger and Mandelbort[5], and Fritchman[6] used Markov chain model; while Elliot[7] and Muntner and Wolf[8] proposed renewal process model. One of the important purposes of channel modeling is to obtain information about the error characteristics of the real channel which could be useful in evaluating the effectiveness of the error control technique and in facilitating the design of an error control system.

II. SIMPLE PARTITIONED MARKOV CHAIN MODEL

(A) A good channel model should meet the following criteria:

(i) It represents the real channel with reasonable accuracy.
(ii) It should be relatively simple.
(iii) Useful information about the channel can be derived from the model.

B. D. Fritchman[6] proposed the simple partitioned Markov chain model. It postulates N-1 error free states and a single error state. Furthermore it postulates that there are no transitions between the error free states. The model can be described by the probability transition matrix.

$$P=\begin{bmatrix} p_{11} & & & & p_{1,N} \\ & p_{22} & & O & p_{2,n} \\ & & \ddots & & \vdots \\ & O & & p_{N-1,N-1} & p_{N-1,N} \\ p_{N1} & p_{N2} & & p_{N,N-1} & p_{NN} \end{bmatrix}$$

(B) The parameters of the model - the transition probabilities can be obtained as follows[9]:

(1) Counting the number of errors of the error data and the number of the gaps, i.e., the error free digits between the errors; construct the gap distribution - the plot of the cumulative relative frequency of the gap of length m versus the length m.

(2) The error-free run distribution is the probability of an error-free run of length at least m following an error, $P(0^m/1)$, versus the length m. The error-free run distribution can be derived from the gap distribution.

(3) Fit the error-free run distribution by a sum of the exponential functions of the form

$$A_1 e^{a_1 m} + A_2 e^{a_2 m} + \text{---} + A_{N-1} e^{a_{N-1} m}$$

The number of terms is dictated by the error-free run distribution.

(4) The transition probabilities are calculated as follows:

$$P_{11} = e^{a_1}, \qquad P_{1N} = 1 - P_{11}, \qquad P_{N1} = A_1 e^{a_1},$$

$$P_{22} = e^{a_2}, \qquad P_{2N} = 1 - P_{22}, \qquad P_{N2} = A_2 e^{a_2},$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$P_{N-1,N-1} = e^{a_{N-1}}, \quad P_{N-1,N} = 1 - P_{N-1,N-1}, \quad P_{N,N-1} = A_{N-1} e^{a_{N-1}},$$

$$P_{NN} = 1 - P_{N1} - P_{N2} \text{ - - - - - } P_{N,N-1}.$$

(5) As an example, error data from a real HF channel is used to obtain the model. The error-free run distirbution is approximated by a sum of two exponential functions

$$A_1 e^{a_1 m} + A_2 e^{a_2 m}$$

where $A_1 = 0.36289620$, $a_1 = -0.00088613597$,
$A_2 = 0.79450446$, $a_2 = -0.30593276$.

The transition probabilities are:

$$p_{11} = e^{a_1} = 0.99911, \quad p_{13} = 1-p_{11} = 0.00089, \quad p_{31} = A_1 e^{a_1} = 0.36258$$

$$p_{22} = e^{a_2} = 0.73644, \quad p_{23} = 1-p_{22} = 0.26356, \quad p_{32} = A_2 e^{a_2} = 0.58510$$

$$p_{33} = 1-p_{31}-p_{32} = 0.05232$$

The transition matrix of the model is

$$\begin{bmatrix} 0.99911 & 0 & 0.00089 \\ 0 & 0.73644 & 0.26356 \\ 0.36258 & 0.58510 & 0.05232 \end{bmatrix}$$

(C) The simple partitioned Markov chain model has only one error state, an error can occur in the error data sequence only when the channel is in the error state. Given that the channel is in the error state the probability that the channel returns to the error state in k steps, i.e.

$$p^{(k)}(N,N)$$

is exactly the first return probability of the error. Let f(k) be the first return probability of the error, i.e. given an error, the probability that the next error occurs after k-1 error free digits

$$1 \overset{1}{\frown} \underbrace{0 \overset{2}{\frown} 0 \overset{3}{\frown} 0 \; --- \; 0 \overset{k-1}{\frown} 0}_{k-1 \ \text{"0"}} \overset{k}{\frown} 1$$

$$f(k) = p^{(k)}(N,N)$$

f(k) can be calculated recurrently from the probability transition matrix [9], [10]. f(k) is a probability density function

$$\sum_{k=1}^{\infty} f(k) = 1$$

Let F(k) be the cumulative first return probability

$$F(k) = \sum_{i=1}^{k} f(i)$$

III. ERROR BURSTS

(A) On real channel errors do not occur independently; in the error sequence, there are more errors in some portions of the sequence than the others. The portions with dense errors are commonly referred to as bursts. In order that the error burst can be uniquely characterized, a specified value of the error density Δ_0 is chosen. With respect to Δ_0 the error burst is defined as follows [11], [12]:

1. (1) A burst begins with an error and ends with an error.
2. (2) The ratio of the number of errors to the number of digits of a burst is larger than or equal to the specified density Δ_0.
3. (3) If successive inclusion of the next error keeps the error density above Δ_0 or equal to Δ_0, the burst is continued; otherwise the burst ends.
4. (4) A burst cannot begin with an error belonging to the previous burst.
5. (5) Finally a single error is a burst of length 1 with 1 error if the inclusion of the next error reduces the density below Δ_0.

(B) Bursts are characterized by a pair of integers: n, the number of errors in the burst and m, the length of the burst. A burst of length m and with n errors is succeeded by at least $[{}^{(n+1)}/\Delta_0] - m$ corrected digits or "0", where $[x] = x$ if x is an integer and $[x]$ is the smallest integer larger than x if x is not an integer.

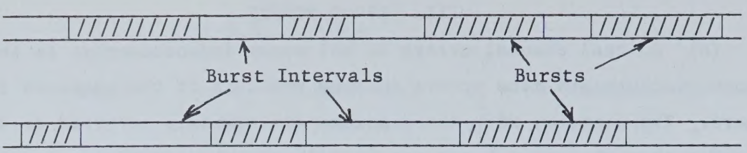
Let m be the length of a sequence of digits, n the number of errors in the sequence, and m_i the length up to i-th error. $m_i = 1$ means that the first digit is an error, $m_n = m$ means that the last digit is an error, and ${}^{i}/m_i \geq \Delta_0$ means that the error density up to i-th error is larger than or equal to the specified value Δ_0. A burst of length m and with n errors is a sequence of m digits satisfying the following:

(i) $m_1 = 1$ (ii) $m_n = m$

(iii) $i/m_i \geq \Delta_0$, for $1 \leq i \leq n$ (iv) ${}^{(n+1)}/m_{n+1} < \Delta_0$

IV. PROBABILITY OF THE BURST

(A) When a specified value of the error density, Δ_0, is chosen, the errors are grouped uniquely into bursts separated by burst intervals - sequences of "0".



Bursts of length m and with n errors form a set. Sets with different values of n and m are disjoint. The probability of burst of length m and with n errors, $P_n(m)$, can be calculated recurrently as follows:

$$P_n(m) = P(n,m)\ [1-F([\frac{n+1}{\Delta_0}] - m)]$$

where

$$P(n,m) = \sum_{\mu=n-1}^{\min([\frac{n-1}{\Delta_0}],\ m-1)} P(n-1,\mu)f(m-\mu), \text{ for } m = n,n+1,\text{--},[\frac{n}{\Delta_0}],$$

and

$$p(2,m) = f(m-1), \quad \text{for } m = 2,3,\ \text{---},\ [\frac{2}{\Delta_0}].$$

And

$$P_1(1) = 1 - F\ ([\frac{2}{\Delta_0}] - 1).$$

(B) The sum of $P_n(m)$ with all possible values of n and m is equal to unity, where the possible values of n and m are:

(i) $n = 1, 2, 3, \text{---},$

(ii) $m = 1, \quad \text{for } n = 1,$

(iii) $m = n, n+1, \text{---}, [\frac{n}{\Delta_0}], \quad \text{for } n \geq z.$

$$\sum_{\text{all possible n and m}}\sum P_n(m) = P_1(1) + \sum_{n=2}^{\infty}\sum_{m=n}^{[\frac{}{\Delta_0}]} P_n(m) = 1$$

This assertion can be shown as follows:

(i) With exception of the single-error burst, all bursts can be grouped according to the first two errors

$$\begin{array}{lllll} 1 & 1 & & & \\ 1 & 0 & 1 & & \\ & \vdots & & & \\ \end{array}$$

$$\underbrace{1 \quad 0 \quad \text{---} \quad 0 \quad 1}_{[\frac{2}{\Delta_0}]}$$

(ii) In each group all error combinations can occur after the second error, hence the probability of each group is

$$\begin{array}{l} f(1) \\ f(2) \\ \vdots \\ f([\frac{2}{\Delta_0}] - 1) \end{array}$$

respectively

(iii) The probability of the single-error burst is

$$1 - F([\frac{2}{\Delta_0}] - 1)$$

(iv) Since

$$f(1) + f(2) + --- + f([\frac{2}{\Delta_0}] - 1) + [1 - F([\frac{2}{\Delta_0}] - 1)] = \sum_{i=1}^{\infty} f(i) = 1$$

Therefore

$$\sum \sum P_n(m) = 1$$

(C) Consider the set of all bursts of length m with n errors as a sample point and assign $P_n(m)$ as its probability, all the bursts form a discrete probability space. The mapping from the sample point to a pair of integers (n,m) is a two dimensional random variable with $P_n(m)$ as the probability density function.

For $\Delta_0 = 1/8$, the density function is as follows:

$$\begin{array}{llll|l}
P_1(1) & & & & P_1 \\
 & P_2(2) & P_2(3) & P_2(4)--P_2(16) & P_2 \\
 & & P_3(3) & P_3(4)--P_3(16)--P_3(24) & P_3 \\
 & & & P_4(4)--P_4(16)--P_4(24)--P_4(32) & P_4 \\
 & & & \vdots & \vdots \\
 & & & \quad P_{16}(16)--P_{16}(24)--P_{16}(32)-- & P_{16} \\
 & & & \qquad \vdots & \vdots \\
\hline
P(1) & P(2) & P(3) & P(4)---P(16)---P(24)----P(32)------ &
\end{array}$$

Summing each row, the marginal density function

$$P_n = \sum_m P_n(m)$$

is the probability of burst with n errors; and summing each column, the marginal density function

$$P(m) = \sum_n P_n(m)$$

is the probability of burst of length m.

V. EVALUATION OF ERROR CORRECTING CODES

(A) The probability density function $P_n(m)$ makes it possible to derive the important statistical quantities, e.g., the expected number of errors in a burst and the expected number of bursts in an error sequence; thus makes it possible to evaluate the performance of an error correcting code. The methodology of evaluating the error correcting codes can be stated as follows:

(1) Given an error correcting code choose a suitable value of the error density Δ_0.

(2) With respect to Δ_0, the sets of the error bursts are divided into three categories:

  (i) the first category consists of the sets, some subset of which can be corrected by the code and the probability of the subset can be easily calculated;

  (ii) the second category consists of the sets which cannot be corrected by the code;

  (iii) the third category consists of the sets some subset of which can also be corrected by the code but their probability is so small that they can be neglected in the calculation.

(3) Knowing the expected number of the bursts in the sets belonging to the first category and the probability of those bursts which can be corrected, the performance of the code can then be calculated.

(B) Two examples of evaluating error correcting codes have been illustrated: block code and error burst correcting code. In the case of block code[11], Golay (24, 12, 3) code is used as an example. It was analyzed that most of the bursts with one, two, and three errors and a part of the bursts with four errors can be corrected and their probability can be easily computed. The bit error rate before coding is 1.8×10^{-3} and the bit error rate after coding is 9.4×10^{-4} from calculation in comparison with 7.1×10^{-4} from simulation.

In the case of diffuse code [13], the code can correct any error combination of length $\leq B = 2x$ with guard space $G > 6x + 2$. In the sets of bursts of length $\leq B$, there are bursts which are preceded and succeeded by G or more error free digits. The probability of the subset can be calculated; thus the performance of the code can be evaluated. The comparison between the calculated and the simulated results are as follows:

B	BER (calculated)	BER (simulated)
16	9.1686×10^{-4}	8.5358×10^{-4}
30	6.2330×10^{-4}	6.9280×10^{-4}
40	5.6908×10^{-4}	5.7879×10^{-4}
60	5.9234×10^{-4}	5.4372×10^{-4}
80	6.7034×10^{-4}	5.1156×10^{-4}

VI. CONCLUSIONS

Although HF channel was used as an example in this paper, the simple partitioned Markov chain model can be applied to other real channels; for instance, the troposcatter channel.

It has been demonstrated[9] that the burst distribution derived from the model has good agreement with simulation. Raise the probability transition matrix P to the M-th power, can be used to represent the channel after interleaving of degree M. The burst distribution[14] obtained from the calculation has again good agreement with simulation.

Error correcting codes have rigid structure and these structures are deterministic. The channel characteristics, on the other hand, is probabilistic. To match a deterministic code to a probabilistic channel is difficult to say the least. However given a performance criterion, to design a satisfactory error control scheme should be possible. The two-dimensional probability density function $P_n(m)$ seems to have the potential in supplying the necessary information about the channel to acheive this goal.

REFERENCES

1. E. N. Gilbert, "Capacity of a burst-noise channel", Bell Sys. Tech. J., vol. 39, pp. 1253-1265, September 1960.
2. E. L. Cohen and S. Berkovits, "Exponential distributions in Markov models for communication channels", Information and Control, vol. 13, pp. 134-139, 1968.
3. S. Berkovits and E. L. Cohen, "A Markov chain model for tropospheric scatter links", Int. J. Electronics, vol. 25, no. 5, pp. 485-491, 1969.
4. S. Berkovits and E. L. Cohen, "Some mathematical computations concerning a Markov chain model for communication channels", Int. J. Electronics, vol. 30, no. 2, pp. 133-140, 1971.
5. J. M. Berger and B. Mandelbrot, "A new model for error clustering in telephone circuits", IBM J. Res. and Develop., June 1963.
6. B. D. Fritchman, "A binary channel characterization using partitioned Markov chains", IEEE Trans. on Information Theory, vol. IT-13,

pp. 221-227, April 1967.

7. E. O. Elliott, "A model of the switched telephone network for data communications", Bell Sys. Tech. J. vol. 44, January 1965.
8. M. Muntner and J. K. Wolf, "Predicted performance of error-control technique over real channel", IEEE Trans. on Information Theory, vol. IT-14, pp. 640-650, September 1968.
9. S. Tsai, "Markov characterization of the HF channel", IEEE Trans. on Communication Technology, vol. COM-17, pp. 24-32, February 1969.
10. W. Feller, "An Introduction to Probability Theory and Its Applications", vol. 1, New York: Wiley, 1958.
11. S. Tsai, "Analytic method of evaluating error correcting codes on a real channel", IEEE International Conference on Communications, Conference Record 1972.
12. K. Brayer and O. Cardinale, "Evaluation of error correction block encoding for high-speed HF data", IEEE Trans. on Communication Technology, vol. COM-15, pp. 371-382, June 1967.
13. S. Tsai, "Diffuse code on a real channel", submitted for publication.
14. S. Tsai and P. S. Schmied, "Interleaving and error burst distribution", IEEE Trans. on Communications, vol. COM-20, pp. 291-296, June 1972.

NOTE ON THE BEST MATCH PROBLEM[*]

L. D. RUDOLPH, K. G. MEHROTRA AND H. SCHWARZLANDER
Syracuse University
Syracuse, New York

ABSTRACT

Given a set of data consisting of a specified number of b-digit binary words, and an arbitrary b-digit query word w, a word in the data set is to be found whose Hamming distance from w is minimum. What is the minimum number of memory references needed, on the average, to accomplish this, given a particular memory size? Minsky and Papert [1] conjectured that for this "Best Match Problem", no significant speed-up can be achieved by permitting a large increase in memory redundancy. An algorithm is presented that serves as a counterexample.

I. INTRODUCTION

Suppose that we are given a body of information--we will call it a data set-- in the form of 2^a distinct binary words each b digits in length where $b > a >> 1$. We will suppose that the data set has been chosen at random from among all possible such sets, so that one cannot expect to find much redundant structure within it. The ordered data set requires about $b \cdot 2^a$ bits of binary information for complete description. We will not, however, be interested in the order of the words in the data set. This reduces the amount of information required to store the set to about $(b-a) \cdot 2^a$ bits.

We want a machine that, when presented with an arbitrary b-digit query word w, will exhibit the word $\hat{w}$ in the data set which is closest to w. ("Best Match Problem"). "Closeness" is measured by Hamming distance,

$$d(w,\hat{w}) = \sum_{i=1}^{b} |x_i - \hat{x}_i|, \quad \text{where } w = (x_1,\ldots,x_b),\ \hat{w} = (\hat{x}_1,\ldots,\hat{x}_b).$$

Thus, the Best Match Problem requires that, given w, we find a $\hat{w}$ in the data set that minimizes $d(w,\hat{w})$.

It is necessary to formulate constraints upon how this machine works in such a way that computational aspects can be separated from memory aspects. To this end, we adopt the scheme which was introduced by Minsky and Papert. We will allow our machine a memory of M binary storage locations, that is, M one-digit binary words. We are required to compose in advance, before we see the data set, two algorithms A_{file} and A_{find} that satisfy the following conditions:

1. A_{file} operates on the data set and fills the M bits of memory with information. Neither the data set nor A_{file} are used again, nor is A_{find} allowed to get any information about what A_{file} did, except by inspecting the contents of the memory.
2. A_{find} is then given a query word, w, and asked to identify the word $\hat{w}$, using the information stored in the memory by A_{file}. We are interested in how many bits of memory A_{find} has to consult in the process.

---

[*] This work was supported by Rome Air Development Center under Contract F30602-70-C-0190.

3. The goal is to optimize the design of A_{file} and A_{find} to minimize the number of memory references in the question-answering computation, averaged over all possible query words w.

Let N*(a,b,M) denote the number of bits referenced, averaged over all possible words w, using the best possible A_{file} - A_{find} pair. For specified values of a and b, we would like to be able to plot a curve of N* as a functions of M, which expresses the tradeoff between memory and computational effort.

It is relatively easy to identify the extremes of this time-memory curve. The minimum amount of memory required for the Best Match Problem is roughly $M_{min} = (b-a)\cdot 2^a$ and the corresponding exhaustive search algorithm presumably requires about $N_{max} = (b-a)\cdot 2^a$ bit references. At the other extreme, we have a b-digit word M_w for each possible query word w, with $M_w = \hat{w}$ where $\hat{w}$ is a word in the data set closest to w. For a given w, it is necessary only to look up M_w and read out $\hat{w}$ which requires b bit references. Hence $M_{max} = b\cdot 2^b$ and $N_{min} = b$. Except for the two extremal points, no simple results are obtainable. It seems clear that for the Best Match Problem, small amounts of memory redundancy are not going to cause a drastic reduction in the number of bit references required.

Yet, the picture is not entirely gloomy, as we now present an A_{file} - A_{find} pair that achieves a significant reduction, as compared to an exhaustive search, in the number of bit references required for the Best Match Problem. The amount of memory required is quite large compared to M_{min}, but is also quite small compared to M_{max}.

II. DESCRIPTION OF THE ALGORITHM

The data set can be thought of as 2^a points chosen at random from a space of 2^b points. Distance in this space is measured according to the Hamming metric. A (Hamming) sphere of radius t and center c is the set of all points at distance t or less from c. There are

$$S(b,t) \equiv 1 + b + \binom{b}{2} + \ldots + \binom{b}{t} \qquad (1)$$

such points. (Note that $S(b,b)=2^b$.) Since there are 2^b points in the space, it is conceivable that we could pack $2^b/S(b,t)$ spheres into the space in such a way that each point is contained in one and only one sphere, i.e., that the spheres fill up the space without overlapping. For certain values of b and t this is possible (e.g., b=23 and t=3), but usually the spheres do not fit together exactly and a perfect packing can only be approximated. Since we only seek order-of-magnitude results, however, we will pretend that a perfect packing is possible for all values of b and t. So let us assume that the space of 2^b points has been partitioned into $2^b/S(b,t)$ spheres of radius t, where c_i is the center of the i^{th} sphere. To the i^{th} sphere we assign a memory location L_i. The number of bits at L_i is left unspecified. The partition into spheres and assignment of memory locations are of course done prior to seeing the data set.

We can now give an informal description of the A_{file} algorithm. Let D_i be the distance from c_i to a nearest word in the data set. Then in location L_i store those data words whose distance from c_i is no greater than D_i+2t. After this has been accomplished for $i=1,2,\ldots,2^b/S(b,t)$, the data set and A_{file} are never used again.

The A_{find} algorithm operates as follows: Given a query word w, find the i such that w is contained in the i^{th} sphere. Then determine, by exhaustive comparison, which data word at location L_i is closest to w. The resulting data word is $\hat{w}$.

That this A_{file} - A_{find} pair always gives a correct result is guaranteed by the triangle inequality for the Hamming metric. Suppose w is in the i^{th} sphere, α is a data word closest to c_i and β is a data word closest to w. By the triangle inequality, we have

$$d(\beta,c_i) \leq d(\beta,w) + d(w,c_i) \quad \text{and} \quad d(w,\alpha) \leq d(w,c_i) + d(c_i,\alpha).$$

Since β is a word closest to w, we also have $d(\beta,w) \leq d(w,\alpha)$. Combining these inequalities gives

$$d(\beta,c_i) \leq d(c_i,\alpha) + 2d(w,c_i).$$

But $d(c_i,\alpha) = D_i$ and $d(w,c_i) \leq t$. Hence

$$d(\beta,c_i) \leq D_i + 2t \tag{2}$$

which means that β is one of the data words stored at location L_i by A_{file}.

We remark here that we are attempting to exploit the distribution of distance between two randomly selected points in a high-dimensional space. The distance between an arbitrary word c_i in the space and all possible words of length b is binomially distributed. The distance between c_i and the words in the data set can also be assumed to be binomially distributed. For large a and b, this means that 'almost all' of the data words will be close to distance b/2 from c_i. However, the distance D_i to the nearest data word will, on the average, be considerably less. The hope is that the expected number of data words in a sphere of radius D_i+2t centered at c_i (the points which A_{file} stores at location L_i) will be small. This turns out to be the case if we choose the radius t to be small enough.

III. ANALYSIS

We now give exact and approximate formulas for the expected memory size M and expected number of bit references N as a function of a,b and t, a time-memory curve for b=100, and asymptotic results. Note that t is a parameter that traces out a time-memory curve as it varies over the range $0 \leq t \leq b$. At t=0, there is a sphere of radius 0 centered at each of the 2^b points in the space, and we need store only one data point at each location L_i. At this extreme, A_{find} becomes a table look-up algorithm. At t=b, there is only one sphere, containing all the data points, and we are forced to compare w with every point in the data set. At this extreme, A_{find} becomes an exhaustive search.

E(M), the expected size of the memory, and E(N), the expected number of bit references, for given t, are given by

$$E(M) = \frac{b2^a}{S(b,t)} \sum_{d_o=0}^{b} \left\{ \left[\sum_{x=d_o}^{b} \binom{b}{x}/2^b \right]^{2^a} - \left[\sum_{x=d_o+1}^{b} \binom{b}{x}/2^b \right]^{2^a} \right\} S(b,d_o+2t), \tag{3}$$

$$E(N) = 2^{-b}S(b,t)E(M). \tag{4}$$

These formulas are derived in the Appendix, along with approximations which were used for actual computations.

A time-menory curve for b=100 and a=20 is shown in Fig. 1. One characteristic of this curve, and of similar ones which have been computed [2], is a sharp drop-off in the expected number of bit references when E(M) exceeds a certain value. It is of interest to see what happens to this threshold as the length of the data word and the size of the data set are increased

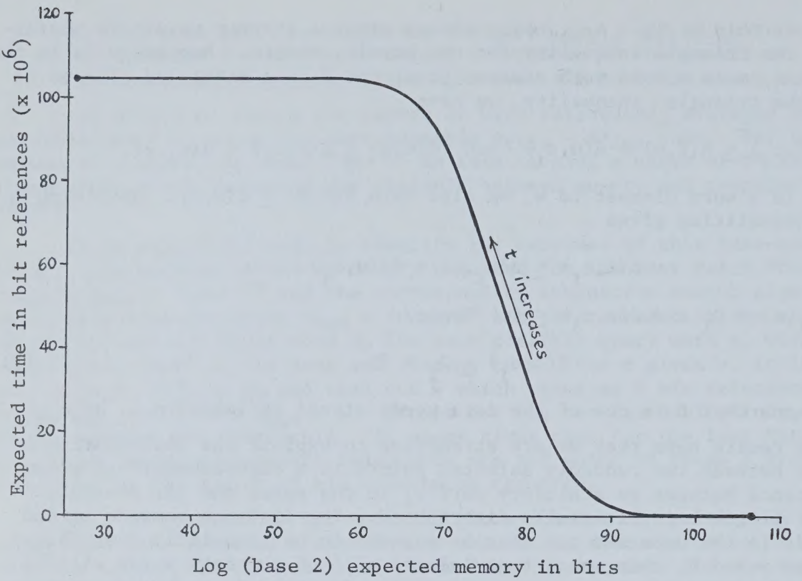


Figure 1. Time vs. memory for b = 100, a = 20

without bound. In order to fix the relative information storage capacities of the data set and the space from which it is selected, we define the parameter

$$r = \frac{\log_2(\text{data set size})}{\log_2(\text{space size})} = \frac{a}{b} \tag{5}$$

which we call the density of the data set. For purposes of obtaining asymptotic results, it is also convenient to define a second dimensionless quantity

$$R = \frac{\log_2[E(M)/M_{min}]}{\log_2[M_{max}/M_{min}]} \tag{6}$$

which we call the memory redundancy. It is shown in the Appendix that for a given density r, the asymptotic time-memory curve is a step function where the step, or threshold, occurs at a memory redundancy of

$$R_c = (1-r)^{-1}[1-H\{1/4 - (1/2)H^{-1}(1-r)\}] \; . \tag{7}$$

where $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function. We call R_c the critical memory redundancy for the algorithm. It is interesting to note here that the location of the threshold relative to M_{min} and M_{max} is asymptotically only a function of the data set density.

IV. IMPLEMENTATION

In our analysis, we ignored the computational complexity of implementing A_{file} and A_{find}. We believe this is justified on the following grounds. First, A_{file} is an incremental rather than a global algorithm. It examines just one member of the data set at a time, with no control over which it will see next, and without any subterfuge of storing the data set internally. Second, A_{find} requires a relatively small amount of time and memory overhead to determine L_i for a given query word w and to carry out the search for the data word stored at location L_i that is closest to w. To justify these assertions, it will be necessary to consider how the algorithm might be implemented.

The first problem is to specify the partition of the space of 2^b points into spheres of radius t, or equivalently, to specify the sphere centers $\{c_i\}$. This sphere-packing problem also occurs in the design of error-correcting codes for the reliable transmission of information through a noisy channel and many efficient and easily specified codes are known. In the coding context, the sphere centers $\{c_i\}$ are the code words and the set of all centers is called a t-error-correcting code of block length b. If the code words were chosen in an arbitrary fashion, the odds are that there would be little or no redundant structure, and the code could only be specified by storing all the code words. Fortunately, it happens that very good sphere-packings can be achieved by codes in which the code words form a k-dimensional subspace of the space of k-tuples over the field of two elements. In this case, the code is called a (b,k) linear code over GF(2). An advantage of a linear code is that it can be specified by storing only k linearly independent code words rather than all 2^k code words. A further simplication is obtained by choosing the (b,k) linear code to be cyclic. In this case, the entire code can be specified by storing only one code word. That good sphere-packings can be achieved through the use of cyclic codes is illustrated by the fact that the b=23, t=3 perfect packing can be obtained by using the well known (23,12) triple-error-correcting Golay cyclic code. Hence, specifying the partition of the space of 2^b points into spheres of radius t can be achieved with an insignificant amount of memory overhead.

The second problem is to determine, given w, which sphere w is in, or equivalently, which sphere center c_i is closest to w. This is just the decoding problem for error-correcting codes in which we think of w as a code word plus an error vector and map (decode) w into the nearest code word c_i. If the sphere-packing is perfect, then the query word w falls in one and only one sphere, and nearest-neighbor decoding yields a unique code word c_i, and from c_i a unique location L_i. If the sphere-packing is not perfect, however, and w does not fall within one of the spheres of radius t, the decoding procedure may yield more than one "nearest code word." In this case, it would be necessary to search the contents of more than one location.

While the encoding (specification) of a linear block code is very simple, the decoding process, which is inherently nonlinear regardless of whether or not the code is linear, is in general quite complex. Fortunately, a code with block length on the order of b=100 is relatively easy to decode, and even for much larger block lengths, certain classes of codes are known that produce relatively good sphere-packings and are easy to decode. Thus, although the decoding of linear block codes is a difficult problem in general, we find that the decoding art has progressed to the point where A_{find} algorithms for data sets of the size considered here could be implemented with relatively modest amounts of time and memory overhead.

APPENDIX

1. A Basic Result.

Suppose A is an m×n matrix whose elements are mutually independent random variables which take value 0 or 1 each with probability 1/2. Let X_i be the weight of the i^{th} row, i.e., the number of 1's in the i^{th} row. Let $W=\min X_i$. Suppose t is a positive integer and K_t is the number of rows of A of weight W+t or less. E(Y) denotes the expected value of a random variable Y.

Theorem 1:

$$E(K_t) = m2^{-n} \sum_{w=0}^{n} \left\{ \left[\sum_{x=w}^{n} \binom{n}{x} (1/2)^n \right]^m - \left[\sum_{x=w+1}^{n} \binom{n}{x} (1/2)^n \right]^m \right\} \sum_{i=0}^{w+T} \binom{n}{i}. \qquad (A\cdot 1)$$

Proof: Since the elements of a row are statistically independent binary random variables, each X_i is a binomial random variable with parameters n and 1/2. If p(B) denotes the probability of the event B, then

$$p(X=x) = \binom{n}{x} (1/2)^n, \qquad x = 0, 1, 2, \ldots, n.$$

By definition, $W = \min\limits_{1 \le i \le m} X_i$. Then for $w=0, 1, 2, \ldots, n$,

$$p(W=w) = p(\min_{1 \le i \le m} X_i = w) = p[\min_{1 \le i \le m} X_i \le w] - p[\min_{1 \le i \le m} X_i \le w - 1]$$
$$= \{1 - p(\min_{1 \le i \le m} X_i > w)\} - \{1 - p(\min_{1 \le i \le m} X_i > w - 1)\}.$$

Using the independence of X_i's, the above expression reduces to

$$p(W=w) = \{p(X_i > w-1)\}^m - \{p(X_i > w)\}^m$$
$$= \left[\sum_{x=w}^{n} \binom{n}{x} (1/2)^n \right]^m - \left[\sum_{x=w+1}^{n} \binom{n}{x} (1/2)^n \right]^m . \qquad (A\cdot 2)$$

Next, the probability that a randomly chosen row will have weight (w+t) or less is $\sum_{i=0}^{w+t} \binom{n}{i} (1/2)^n$. Define

$$U_i = \left\{ \begin{array}{l} 1 \text{ if the weight of the } i^{\text{th}} \text{ row} \le W+t \\ 0 \text{ otherwise} \end{array} \right\} \quad i = 1,2,\ldots,m.$$

$$\text{Then } K_t = \sum_{i=1}^{m} U_i, \text{ and } E(K_t) = E\left[\sum_{i=1}^{m} U_i \right] = \sum_{i=1}^{m} E(U_i) = mE(U_1), \qquad (A\cdot 3)$$

by symmetry. Also,

$$E(U_1) = p(U_1=1) = \sum_{w=0}^{n} p(U_1=1 | W=w) \cdot p(W=w) = \sum_{w=0}^{n} \left[\sum_{i=0}^{w+t} \binom{n}{i} (1/2)^n \right] p(W=w).$$

Substituting this value of $E(U_1)$ in (A·3) and the value of p(W=w) given by (A·2), the theorem is proved.

Now, set $a_w = \left[\sum_{x=w}^{n} \binom{n}{x} (1/2)^n \right]^m$.

Clearly $a_0 = 1$ and $a_{n+1} = 0$. Then

$$m^{-1} 2^n E(K_t) = \sum_{w=0}^{n} \{a_w - a_{w+1}\} \sum_{i=0}^{w+t} \binom{n}{i} = a_0 \sum_{i=0}^{t} \binom{n}{i} + \sum_{w=1}^{n} a_w \binom{n}{w+t}$$
$$= a_0 \sum_{i=0}^{t} \binom{n}{i} + \sum_{w=1}^{n-t} a_w \binom{n}{w+t},$$

since, by convention, $\binom{n}{i} = 0$ for i>n. Substituting the value of a_w, we obtain

$$m^{-1}2^n\, E(K_t) = \sum_{i=0}^{t} \binom{n}{i} + \sum_{w=1}^{n-t} \left[\sum_{x=w}^{n} \binom{n}{x} (1/2)^n \right]^m \binom{n}{w+t}$$

$$= \sum_{i=0}^{t} \binom{n}{i} + \sum_{w=1}^{n-t} \left[1 - \sum_{x=0}^{w-1} \binom{n}{x} (1/2)^n \right]^m \binom{n}{w+t}$$

Now we are ready to find an approximation for $E(K_t)$. We use the fact that

$$\left(1 - \frac{\alpha}{m}\right)^m \approx e^{-\alpha} \text{ for large values of } m.$$

Writing

$$1 - \sum_{x=0}^{w-1} \binom{n}{x} (1/2)^n = 1 - \frac{m \sum_{x=0}^{w-1} \binom{n}{x} (1/2)^n}{m},$$

we identify the numerator of the second term on the right with α. Thus,

$$m^{-1}2^n\, E(K_t) \approx \sum_{i=0}^{t} \binom{n}{i} + \sum_{w=1}^{n-t} \binom{n}{w+t} \exp \{-m \sum_{x=0}^{w-1} \binom{n}{x} (1/2)^n\}. \qquad (A{\cdot}4)$$

2. Expected Memory and Number of Bit References

The space of 2^b words is assumed to be partitioned into $m = 2^b/S(b,t)$ non-overlapping spheres of Hamming radius t and centers $\{c_i\}$. Let D_i denote the distance from c_i to a nearest data word. Then A_{file} stores at location L_i all data words that are distances $D_i + 2t$ or less from c_i. Our first problem is to find the expected number of data words stored at L_i.

For a given data set, let $K_t^{(i)}$ denote the number of data words stored at location L_i, $i=1, 2,\dots,m$. Because a data set is assumed to be selected randomly from the space, and because of inherent symmetry, $E(K_t^{(1)}) = E(K_t^{(2)}) = \dots = E(K_t^{(m)})$.

Without loss of generality, we can choose the center of a representative sphere to be c_0, the all-0 b-tuple. The distance between c_0 and any data word is then the Hamming weight of (number of 1's in) the data word (i.e., d_0 is a random variable expressing the weight of the lowest-weight word in the data set). Then we wish to evaluate the expected number of words in the data set of weight $d_0 + 2t$ or less. From Theorem 1, this is (using $n=b$, $m=2^a$):

$$E(K_{2t}) = 2^{-(b-a)} \sum_{d_0=0}^{b} \left\{ \left[\sum_{x=d_0}^{n} \binom{b}{x} (1/2)^b \right]^{2^a} - \left[\sum_{x=d_0+1}^{b} \binom{b}{x} (1/2)^b \right]^{2^a} \right\} S(b,d_0+2t), \qquad (A{\cdot}5)$$

or by (A·4)

$$E(K_{2t}) \approx 2^{-(b-a)} \left\{ S(b,2t) + \sum_{d_0=1}^{b-2t} \binom{b}{d_0+2t} \exp\left[-2^{-(b-a)} S(b,d_0-1)\right] \right\}. \qquad (A{\cdot}6)$$

The total memory given by M = [(number of locations) x (average number of words per location) x (bits per word)] has the following expected value:

$$E(M) = \frac{2^b}{S(b,t)} \cdot b \cdot E(K_{2t}). \qquad (A{\cdot}7)$$

Further, the expected number of bit references $E(N)$ takes the value

$$E(N) = b \cdot E(K_{2t}). \qquad (A{\cdot}8)$$

3. Asymptotic Threshold.

In this section we consider the asymptotic behavior of the expected memory as the word length, b, approaches infinity. We assume that the size of the collection of data words, 2^a, also increases at a rate determined by the relation $a = br$, for some fixed r, $0 \leq r \leq 1$. First we consider the limiting distribution of W, for $m = 2^{br}$ and $n = b$, as $b \to \infty$.

Consider $0<r<1$. From (A·2),

$$P(w) \equiv p(W \leq w) = \sum_{x=0}^{w} p(W=x) = 1 - \left[1 - \sum_{0}^{w} \binom{b}{x} 2^{-b}\right]^{2^{br}} . \qquad (A\cdot 9)$$

For any fixed w, as $b \to \infty$, $\sum_{x=0}^{w} \binom{b}{x} 2^{-b}\ 2^{br} \to 0$ implying that $P(w) \to 1$.

Instead, we take $w = b\alpha\,(0<\alpha<1)$ and we will be interested in that value of α for which $P(w)$ changes from 0 to 1.

Assume that for each b we can find a $\beta = \beta(\alpha)$ such that $S(b,b\alpha)=2^{b\beta}$. Then, from (A·9) and the above equality after replacing w by b and taking the natural logarithm, we obtain

$$\ln[1-P(b\alpha)] = 2^{br} \ln\{1 - 2^{-b(1-\beta)}\}.$$

Expanding the right-hand side for $0 \leq \beta < 1$ gives

$$\lim_{b\to\infty} \ln[1-P(b\alpha)] = \lim_{b\to\infty} 2^{br} \{-2^{-b(1-\beta)} - \frac{2^{-2b(1-\beta)}}{2} - \frac{2^{-3b(1-\beta)}}{3} \ldots\}$$

$$= \lim_{b\to\infty} \{-2^{b[r+\beta-1]} - \frac{1}{2}\, 2^{b[r+2\beta-2]} - \frac{1}{3}\, 2^{b[r+3\beta-3]} \ldots\}$$

$$= \begin{cases} -\infty & > \\ -1 \quad \text{if } \beta & = 1-r. \\ 0 & < \end{cases}$$

$$\text{Thus, } \lim_{b\to\infty} P(b\alpha) = \begin{cases} 0 & \text{if } \beta < 1-r \\ 1-1/e & \beta = 1-r \\ 1 & \beta > 1-r \end{cases}$$

Hence,

Theorem 2: In the limit, the random variable W/b takes value α with probability 1 and all other values in the interval [0,1] with probability 0, where α satisfies

$$S(b,b\alpha) = 2^{(1-r)b} \qquad (A\cdot 10)$$

Remark 1: Let us observe that, in the special case when r=1, $1-r = 0$ so that $\beta \geq 0$ and $S(b,b\alpha) = 2^{0\cdot b} = 1$ is satisfied only for $\alpha = 0$, thus implying that W takes value 0 with probability 1 and all other values with probability 0.

At the other extreme, r = 0, we have only one point in the data set and the minimum weight in this set is the weight of this one word. Consequently, the distribution will continue to be binomial with increasing value of b, with expected value b/2. A similar argument seems to hold for a very small neighborhood consisting of $0 < r < 1/b$. In what follows, we will restrict r to the range $1/b < r < 1$.

The sum of consecutive binomial coefficients can be approximated by the binary entropy function H as follows:

$$S(b,b\alpha) \approx 2^{bH(\alpha)} \qquad (A\cdot 11)$$

Thus, by (A·10), $2^{bH(\alpha)} \approx 2^{b(1-r)}$, or $H(\alpha) \cong (1-r)$, or $\alpha \cong H^{-1}(1-r)$ (A·12)

Remark: The function H^{-1} is not well defined in the full range because $H(x)$ is a 2-1 function. But, in the problem under consideration, α lies only in the interval $(0,1/2)$. Thus, in equation (A·11) that value of α is chosen which lies in $(0,1/2)$, making it unique.

By Theorem 2, an approximation of the type (A·11), and using (A·12) in (A·7), the asymptotic expression for the expected memory requirement is given by

$$E(M) \approx b\ 2^{b} \cdot 2^{-bH(\frac{t}{b})}\ 2^{-b(1-r)}\ 2^{bH[2\frac{t}{b} + H^{-1}(1-r)]}$$

$$\approx b\ 2^{b[r+H\{2\frac{t}{b} + H^{-1}(1-r\} - H(\frac{t}{b})]} \qquad (A\cdot 13)$$

Similarly, from (A·8)

$$E(N) \approx b\ 2^{b\{H[2\frac{t}{b} + H^{-1}(1-r)] + r-1\}} \qquad (A\cdot 14)$$

From the above expression, it can easily be seen that for a fixed b and r the maximum value of $E(N)$ occurs at $t = t_0$ where

$$2\frac{t_0}{b} + H^{-1}(1-r) = \frac{1}{2}$$

$$\text{or} \quad t_0 = b(\tfrac{1}{4} - \tfrac{1}{2} H^{-1}(1-r)), \qquad (A\cdot 15)$$

and a sharp decrease is observed in the value at $t = t_0 - 1$.

Using the above result, asymptotically, the expected memory at the threshold point is given by

$$E(M) \approx b\ 2^{b[1+r-H\{1/4 - 1/2\ H^{-1}(1-r)\}]}. \qquad (A\cdot 16)$$

Recall that the minimum and maximum possible memories are respectively given by

$$M_{min} = (b-a)\ 2^{a} = b(1-r)\ 2^{br} \text{ and } M_{max} = b\ 2^{b}.$$

Therefore, the asymptotic memory redundancy, from (6), is given by

$$\lim_{b\to\infty} R = \lim_{b\to\infty} \frac{b[1-H\{1/4 - 1/2\ H^{-1}(1-r)]\} - \log_2(1-r)}{b[1-r] - \log_2(1-r)}$$

$$= (1-r)^{-1}[1-H\{1/4 - 1/2\ H^{-1}(1-r)\}]. \qquad (A\cdot 17)$$

REFERENCES

[1] M. Minsky and S. Papert, Perceptrons, an Introduction to Computational Geometry, M.I.T. Press, Cambridge, Mass., 1969.

[2] L. D. Rudolph, K. G. Mehrotra, and R. J. Longobardi, "Some Results on the Best Match Problem," System and Information Science Report, Syracuse University, Syracuse, N.Y., February, 1972.

UNIVERSAL TESSELLATION AUTOMATA: AN APPLICATION OF COMBINATORY LOGIC TO AUTOMATA THEORY*

WILLIAM I. GROSKY
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

In this study, we show the existence of numerous infinite state tessellation automata, each one of which can simulate a large class of other tessellation automata, including virtually all those in which people have had a theoretical interest. We also show that many constructions of initial configurations for these tessellation automata, whose evolution through time are meant to embody some calculation or construction, can be done in an effective manner. These results are accomplished by using various n-dimensional generalizations of pure combinatory logic.

INTRODUCTION

All recent theoretical studies of tessellation automata - for example, those of Amoroso and Cooper (1,2), Ostrand (3), Smith (4,5,6), and Yamada and Amoroso (7) - have been concerned with those particular finite state automata in which the neighborhood structure is spatially independent and fixed for all time, and in which the choice of the local state transition function to be applied to a particular cell is independent of the position of that cell in the array, though it may be temporally dependent. As Yamada and Amoroso (7) point out, it might prove of interest to consider tessellation automata in which the above properties do not hold. Thus, in this study, we address ourselves to the question of whether any advantages and/or insights accrue from formulating a not necessarily finite state tessellation automaton in which the neighborhood structure of, and the choice of the local state transition function to be applied to, a given cell at a particular time, depends on the total configuration at that time and the relative position of the cell in the array.

We have answered this question in the affirmative by formulating numerous individual, locally controlled tessellation automata of this latter type, each one of which can simulate a wide class of other tessellation automata, including all those with the above uniformity properties. Thus, we have found the existence of many tessellation automata, each one of which can be said to be universal with respect to a large class of other tessellation automata, including virtually all those in which people have had a theoretical interest. For these universal tessellation automata, we have also demonstrated the fact that numerous constructions of initial configurations, whose evolution through time are meant to embody some calculation or construction, can be done in an effective manner, which is certainly not the case with those automata previously studied. Thus, instead of being given a tessellation automaton and a procedure, and asking oneself whether the former is capable of carrying out the latter, we have, in many cases, a method for effectively constructing an initial configuration for one of our newly formulated tessellation automata, whose evolution through time embodies the execution of this procedure. Hence, these newly formulated tessellation automata are also universal in the sense of not being procedure bound; that is, not being overly sensitive to the type of procedure which it is to carry out. This is not to say that previously formulated

---

* This work was supported in part by National Science Foundation Grant GN-655

tessellation automata are all procedure bound, but, as in the investigations of Amoroso and Cooper (2) and Ostrand (3), some tessellation automata were designed to accomplish one task - such as the reproduction of patterns - and it is not clear what else they can do.

In order to be more explicit about our newly formulated tessellation automata, as well as the class of tessellation automata which they can simulate, we must exhibit some definitions. The main definition, constructed in the spirit of Yamada and Amoroso (7), is that of a Euclidean tessellation automaton which doesn't necessarily possess the uniformities which previously formulated tessellation automata have had.

GENERALIZED TESSELLATION AUTOMATA

Definition 1: A *generalized n-dimensional tessellation automaton* (g.t.a.), G, is an ordered 5-tuple, $<S,E,n,\alpha,\beta>$, where,

1) S is an arbitrary non-empty set whose elements are called *states*

2) E is an arbitrary non-empty set whose elements are called *external inputs*. The set $INP = \{f \mid f:Z^n \to E\}$ is called the set of *external input configurations* of G.

3) Z^n is the set of all n-tuples of integers. For $\underline{j} \in Z^n$, we refer to $\underline{j}$ as a *cell* of G. The set $CON = \{f \mid f:Z^n \to S\}$ is called the set of *array configurations* of G. For $s \in S$ and $\underline{j} \in Z^n$, we say that cell $\underline{j}$ *contains* s in the array configuration c, if $c(\underline{j}) = s$.

4) $\alpha:Z^n \times CON \times E \to \bigcup_{k \geq 1}(Z^n)^k \cup (Z^n)^\omega$ - where ω is the first infinite ordinal- and is called the *neighborhood determinant* of G. For $\underline{j} \in Z^n$, $c \in CON$, and $e \in E$, let k* be that element of $\omega \cup \{\omega\}$ which is such that $\alpha(\underline{j},c,e) \in (Z^n)^{k*}$. We call k* the *degree of interaction* of cell $\underline{j}$ in array configuration c with respect to external input e, and denote it by '$deg_{\underline{j},c,e}$'. $\alpha(\underline{j},c,e)$ is called the *neighborhood index* of cell $\underline{j}$ in array configuration c with respect to external input e. The element θ of $(Z^n)^{deg_{\underline{j},c,e}}$ is called the *neighborhood* of cell $\underline{j}$ in array configuration c with respect to external input e, if, for each $\zeta \in deg_{\underline{j},c,e}$, we have that $\theta(\zeta) = \underline{j} + \alpha(\underline{j},c,e)(\zeta)$.

5) $\beta:Z^n \times CON \times E \to \bigcup_{k \geq 1}\{f \mid f:S^k \to S\} \cup \{f \mid f:S^\omega \to S\}$, and is such that $\beta(\underline{j},c,e) \in \{f \mid f:S^{deg_{\underline{j},c,e}} \to S\}$. β is called the *transition determinant* of G, and $\beta(\underline{j},c,e)$ is called the *local state transition function* at cell $\underline{j}$ in array configuration c with respect to external input e.

Now, α, β, and each element $\gamma \in INP$, determine an element of CON^{CON}, called a *parallel transformation*, and denoted by 'τ_γ', which is such that $\tau_\gamma(c) = c'$, if, for each cell $\underline{j} \in Z^n$, $c'(\underline{j}) = \beta(\underline{j},c,\gamma(\underline{j}))(\Gamma)$, where Γ is that element of $S^{deg_{\underline{j},c,\gamma(\underline{j})}}$ which is such that, for each $\zeta \in deg_{\underline{j},c,\gamma(\underline{j})}$, we have $\Gamma(\zeta) = c(\underline{j} + \alpha(\underline{j},c,\gamma(\underline{j}))(\zeta))$.

The set P_G, of *processes* of G, is that subset of CON^ω which is such that, if $p \in P_G$, then, for each $m \geq 0$, $p(m+1) = \tau_{\Psi(m)}(p(m))$, for some $\Psi \in INP^\omega$, called an *external input history*. We say that Ψ is the external input history *corresponding* to p and that p is the process *determined* by Ψ and p(0).

Definition 2: We say that $G = \langle S, E, n, \alpha, \beta \rangle$ is,

1) of *state size* μ, if $\mathrm{card}(S) = \mu$
2) *finitely variant in neighborhood structure* (f.v.n.s.), if, for each cell $\underline{j} \in Z^n$ and each $c \in \mathrm{CON}$, we have $\mathrm{card}(\{\alpha(\underline{j},c,e) \mid e \in E\})$ is finite
3) *finitely variant with respect to transitions* (f.v.t.), if, for each cell $\underline{j} \in Z^n$ and each $c \in \mathrm{CON}$, we have $\mathrm{card}(\{\beta(\underline{j},c,e) \mid e \in E\})$ is finite
4) of *neighborhood bound* δ, for $\delta \in \omega \cup \{\omega\}$, if, for each cell $\underline{j} \in Z^n$, each $c \in \mathrm{CON}$, and each $e \in E$, we have $\deg_{\underline{j},c,e} \in \delta \cup \{\delta\}$, and there is some cell $\underline{j}' \in Z^n$, some $c' \in \mathrm{CON}$ and some $e' \in E$ such that $\deg_{\underline{j}',c',e'} = \delta$
5) *spatially uniform in neighborhood structure* (s.u.n.s.), if α is independent of its first argument
6) *spatially uniform with respect to transitions* (s.u.t.), if β is independent of its first argument
7) *configurationally uniform in neighborhood structure* (c.u.n.s.), if α is independent of its second argument
8) *configurationally uniform with respect to transitions* (c.u.t.), if β is independent of its second argument
9) *neighborhood deterministic* (n.d.), if α is independent of its third argument
10) *transition deterministic* (t.d.), if β is independent of its third argument
11) *transitionally effective* (t.e.), if it is of neighborhood bound $\delta \in \omega$, and, for each cell $\underline{j} \in Z^n$, each $c \in \mathrm{CON}$ and each $e \in E$, $\beta(\underline{j},c,e)$ is effectively computable

The preceding formulation of a tessellation automaton includes as special cases, to the best of our knowledge, all other Euclidean formulations which have appeared heretofore in the literature. This formulation, in its full generality, might be said to be lacking in content, in that an appropriate g.t.a. can easily be constructed to realize any process, as in the following,

Lemma 1: Suppose $S \neq \emptyset$, $\mathrm{card}(S) \leq \mathrm{card}(E)$, and,

$$\alpha: Z^n \times \{f \mid f: Z^n \to S\} \times E \to \bigcup_{k \geq 1} (Z^n)^k \cup (Z^n)^\omega$$

Then, there exists and n-dimensional g.t.a., $G = \langle S, E, n, \alpha, \beta \rangle$, which is such that $P_G = \mathrm{CON}^\omega$.

Proof: Since $\mathrm{card}(S) \leq \mathrm{card}(E)$, there exists a one-one function from S onto a subset of E. Let θ be such a function. We now define,

$$\beta: Z^n \times \{f \mid f: Z^n \to S\} \times E \to \bigcup_{k \geq 1} \{f \mid f: S^k \to S\} \cup \{f \mid f: S^\omega \to S\},$$

as follows,

1) For $\underline{j} \in Z^n$, $c \in \{f \mid f: Z^n \to S\}$, $e \in E - \mathrm{range}(\theta)$, we let $\beta(\underline{j},c,e)$ be arbitrary
2) For $\underline{j} \in Z^n$, $c \in \{f \mid f: Z^n \to S\}$, $e \in \mathrm{range}(\theta)$, we let, $\beta(\underline{j},c,e): S^{\alpha(\underline{j},c,e)} \to \{s\}$, where $s \in S$, $\theta(s) = e$

Suppose $p \in \mathrm{CON}^\omega$. We define an external input history $\Psi_p \in \mathrm{INP}^\omega$ by, $\Psi_p(m)(\underline{j}) = \theta(p(m+1)(\underline{j}))$, for $m \geq 0$, $\underline{j} \in Z^n$. It is easily seen that p is the process of G determined by Ψ_p and $p(0)$, and thus, that $p \in P_G$. Hence, we have that $\mathrm{CON}^\omega \subseteq P_G$. The converse follows from the definition of P_G. QED

The point, though, is not that this formulation is so all-encompassing that it might be thought to be vacuous, but that it unifies all previously done work under a common methodological framework, in which various constraints may be introduced, as exemplified by the preceding eleven definitions, and the resulting behavior examined. The value of the majority of these constraints would be directly related to the realizability of the resulting constrained g.t.a. as a real-world model for computation and construction. Those constraints not so realizable would be of purely theoretical interest, but of interest nonetheless. As an example, consider the following,

Definition 3 (see Smith (4)): Let $G = \langle S, E, n, \alpha, \beta \rangle$ and $G^* = \langle S^*, E^*, n, \alpha^*, \beta^* \rangle$ be two n-dimensional g.t.a.'s. For $q, r \geq 1$, we say that *G^* simulates G in r/q times real time*, if there are effectively computable injective mappings $\Delta : CON \rightarrow CON^*$ and $\Gamma : INP^q \rightarrow INP^{*r}$, such that, for any $c \in CON$ and $\langle \gamma_1, \ldots, \gamma_q \rangle \in INP^q$, we have,

$$\Delta(\tau_{\gamma_q}(\ldots(\tau_{\gamma_1}(c))\ldots)) = \tau^*_{\gamma^*_r}(\ldots(\tau^*_{\gamma^*_1}(\Delta(c)))\ldots),$$

where $\langle \gamma^*_1, \ldots, \gamma^*_r \rangle = \Gamma(\langle \gamma_1, \ldots, \gamma_q \rangle)$. We call $\langle \Delta, \Gamma \rangle$ the *simulation* under consideration. The above simulation is called *uniform*, if there is an $e^* \in E^*$, such that, for each $\xi \in INP^q$, each $1 \leqslant i \leqslant r$, and each cell $\underline{j} \in Z^n$, we have $\Gamma(\xi)(i)(\underline{j}) = e^*$. If $r = q = 1$, we say that G^* simulates G in *real time*.

Lemma 2: For any n-dimensional g.t.a., G, there is another n-dimensional g.t.a., G*, with the same state set as G, which is s.u.n.s., s.u.t.. c.u.n.s., and c.u.t., and which is such that $P_G \subseteq P_{G^*}$.

Proof: Suppose $G = \langle S, E, n, \alpha, \beta \rangle$. We then let $G^* = \langle S^*, E^*, n, \alpha^*, \beta^* \rangle$, where, $S^* = S$, $E^* = Z^n \times CON \times E$, and where, for $\underline{j} \in Z^n$, $c^* \in CON^* = CON$, and $e^* = \langle \underline{k}, d, e \rangle \in E^*$, we have $\alpha^*(\underline{j}, c^*, e^*) = \alpha(\underline{k}, d, e)$, and $\beta^*(\underline{j}, c^*, e^*) = \beta(\underline{k}, d, e)$. Now, suppose that $p \in P_G$. Thus, p is a process of G determined by some external input history $\Psi_p \in INP^\omega$ and p(0). Consider the external input history $\Psi^*_p \in INP^{*\omega}$ of G* defined by $\Psi^*_p(m)(\underline{k}) = \langle \underline{k}, p(m), \Psi_p(m)(\underline{k}) \rangle$ for all $m \geqslant 0$, $\underline{k} \in Z^n$. It is easily seen that the process of G* determined by Ψ^*_p and p(0) is p. Hence, we get that $P_G \subseteq P_{G^*}$. It is also seen that the other statements of the Lemma immediately follow from the definition of G*.

QED

Lemma 3: For any n-dimensional g.t.a., G, there is another n-dimensional g.t.a., G*, with the same state set as G, which is s.u.n.s. and s.u.t., which is such that $P_G \subseteq P_{G^*}$, and which simulates G in real time.

Proof: Suppose $G = \langle S, E, n, \alpha, \beta \rangle$. We then let $G^* = \langle S^*, E^*, n, \alpha^*, \beta^* \rangle$, where, $S^* = S$, $E^* = Z^n \times E$, and where, for $\underline{j} \in Z^n$, $c^* \in CON^* = CON$, and $e^* = \langle \underline{k}, e \rangle \in E^*$, we have $\alpha^*(\underline{j}, c^*, e^*) = \alpha(\underline{k}, c^*, e)$, and $\beta^*(\underline{j}, c^*, e^*) = \beta(\underline{k}, c^*, e)$. Now, suppose that $p \in P_G$. Thus, p is a process of G determined by some external input history $\Psi_p \in INP^\omega$ and p(0). Consider the external input history $\Psi^*_p \in INP^{*\omega}$ of G* defined by $\Psi^*_p(m)(\underline{k}) = \langle \underline{k}, \Psi_p(m)(\underline{k}) \rangle$ for all $m \geqslant 0$, $\underline{k} \in Z^n$. It is easily seen that the process of G* determined by Ψ^*_p and p(0) is p. Hence, we get that $P_G \subseteq P_{G^*}$. It is also seen that G* simulates G in real time, where we define the simulation $\langle\Delta,\Gamma\rangle$ as follows,

1) $\Delta:\mathrm{CON} \rightarrow \mathrm{CON}^*$ is the identity map

2) $\Gamma:\mathrm{INP} \rightarrow \mathrm{INP}^*$ is such that, for $\gamma \in \mathrm{INP}$, $\Gamma(\gamma)(\underline{j}) = \langle\underline{j},\gamma(\underline{j})\rangle$, for all $\underline{j} \in Z^n$

The rest of the Lemma immediately follows from the definition of G*.

QED

COMBINATORY TESSELLATION AUTOMATA

Introduction and Statement of Main Theorem

The universal g.t.a.'s we formulate in this study will be of denumerable state size, finitely variant in neighborhood structure, of neighborhood bound 2, and neighborhood deterministic. The non-uniformities which do exist, though, are of a particularly simple kind, as will presently be made clear. The class of g.t.a.'s to which these belong is called the class of combinatory g.t.a.'s. The formulation of them utilizes various n-dimensional generalizations of pure combinatory logic - see Curry and Feys (8) and Fitch (9). Just as combinatory logic can be viewed as a syntactic manipulation of strings of symbols, these generalizations of it can be viewed as a syntactic manipulation of arrays of strings of symbols. To be more specific, the state set of a combinatory g.t.a. will consist of some subset of the well-formed formulas (wffs) of some formal system, and the transition from one configuration to another will entail the 'movement' and/or 'combination' of various of these wffs.

Main Theorem: There exists an n-dimensional g.t.a. $G\# = \langle S\#,E\#,n,\alpha\#,\beta\#\rangle$ which is of denumerable state size, finitely variant in neighborhood structure, of neighborhood bound 2, and neighborhood deterministic, and which is such that, if $G = \langle S,E,n,\alpha,\beta\rangle$ is of countable state size, finitely variant in neighborhood structure, finitely variant with respect to transitions, of finite neighborhood bound δ, configurationally uniform in neighborhood structure, configurationally uniform with respect to transitions, and transitionally effective, then G# simulates G in $\delta+3$ times real time. In addition, if G is neighborhood deterministic and transition deterministic, then G# uniformly simulates G in $\delta+2$ times real time.

Construction of G# in Two Dimensions

In this section we will construct G# for dimension $n = 2$, and in the next section we will show that it has the requisite properties. A proof for other values of n is constructed in a similar fashion.

Definition 4: We define the set of primitive symbols SYM = {'B', 'C', 'I', 'K', 'W'} $\cup$ {'$\rho_{i,j}$'| $|i|+|j| \geqslant 1$} $\cup$ {'$\sigma_{i,j}$'| $|i|+|j| \geqslant 1$}. The set of wffs are defined as follows,

1) All primitive symbols are wffs

2) If *X* and *Y* are wffs, then (*XY*) is a wff

3) If *X* is a wff, then it is shown to be so by virtue of 1) and 2)

(Note that we are using italic upper-case letters from the end of the alphabet as meta-variables whose values are wffs. We will also use each primitive symbol as a meta-constant which denotes itself. A string of meta-constants and meta-variables will also be considered as either a meta-constant or a meta-variable, whose values are those wffs which are derived from the given string by replacing each meta-variable by a wff and each meta-constant by its appropriate denotation, taking care that the same wff is substituted for every occurrence of the same meta-variable.)

We sometimes omit parentheses with the convention that grouping is from the left; ie., '*XYZ*' is an abbreviation for '((*XY*)*Z*)', and '*V*(*XY*)*Z*' is an abbreviation for '$((V(XY))Z)$'.

We now define various relations on the set of all wffs,

1) $X >_{\mathrm{B}} Y \leftrightarrow (\exists T,U,V)(X = \mathrm{B}TUV \wedge Y = T(UV))$

2) $X >_{\mathrm{C}} Y \leftrightarrow (\exists T,U,V)(X = \mathrm{C}TUV \wedge Y = TVU)$

3) $X >_{\mathrm{I}} Y \leftrightarrow (\exists T)(X = \mathrm{I}T \wedge Y = T)$

4) $X >_{\mathrm{K}} Y \leftrightarrow (\exists T,U)(X = \mathrm{K}TU \wedge Y = T)$

5) $X >_{\mathrm{W}} Y \leftrightarrow (\exists T,U)(X = \mathrm{W}TU \wedge Y = TUU)$

6) $X >_{*} Y \leftrightarrow (X >_{\mathrm{B}} Y) \vee (X >_{\mathrm{C}} Y) \vee (X >_{\mathrm{I}} Y) \vee (X >_{\mathrm{K}} Y) \vee (X >_{\mathrm{W}} Y) \vee$
$(\exists T,U,V)(T >_{*} U \wedge X = TV \wedge Y = UV) \vee$
$(\exists T,U,V)(T >_{*} U \wedge X = VT \wedge Y = VU)$

7) $X > Y \leftrightarrow (X >_{*} Y) \vee (\exists Z)(X > Z \wedge Z >_{*} Y)$

8) $X \geqslant Y \leftrightarrow (X > Y) \vee (X = Y)$

If $X >_{\mathrm{B}} Y$, we say that X is a *direct B-expansion* of Y, and that Y is a *direct B-reduction* of X. Similarly for $>_{\mathrm{C}}$, $>_{\mathrm{I}}$, $>_{\mathrm{K}}$, and $>_{\mathrm{W}}$. If $X >_{*} Y$, we say that X is a *direct expansion* of Y, and that Y is a *direct reduction* of X. If $X > Y$, we say that X is a *proper expansion* of Y, and that Y is a *proper reduction* of X. If $X \geqslant Y$, we say that X is an *expansion* of Y, and that Y is a *reduction* of X. A wff which has no direct reduction is called *irreducible*.

Suppose that $X \geqslant Y$ and Y is irreducible. Then Y is said to be the *normal form* of X. Not all wffs possess normal forms, as, for example, the wff '$\mathrm{WI}(\mathrm{WI})$'.

We now define the state set S# of G# as consisting of all irreducible wffs. We let the set of external inputs E# of G# be the set of wffs $\{\mathrm{I},\mathrm{CI}\underline{0}, \mathrm{CI}\underline{1},\mathrm{CI}\underline{2},\ldots\}$, where, for $n \geqslant 0$, the wff $\underline{n}$ is a particular irreducible wff which represents the integer n - see Fitch (9). (The wff $\underline{n}$ is such that, for any two wffs X and Y, we have that $\underline{n}XY > X\underbrace{(X(\ldots(XY)}_{n}\ldots))$.)

We now define the transition behavior of G#,

Suppose cell (j,k) ε Z^2 contains $\rho_{r,s}X_1\ldots X_m$ in the array configuration c, for some $m \geqslant 1$. Then, we say that cell (j,k) is in an *active transmission mode to cell (j+r,k+s) with respect to array configuration c*, while cell (j+r,k+s) is in a *passive acceptance mode from cell (j,k) with respect to array configuration c*. We also call X_1 the *active message from cell (j,k) to cell (j+r,k+s) with respect to array configuration c*. Suppose cell (j,k) contains either $\sigma_{r,s}$ or $\sigma_{r,s}X_1\ldots X_m$ in the array configuration c, for some $m \geqslant 1$. Then, we say that cell (j,k) is in an *active acceptance mode from cell (j+r,k+s) with respect to array configuration c*, while cell (j+r,k+s) is in a *passive transmission mode to cell (j,k) with respect to array configuration c*. In this case, we define the *passive message from cell (j+r,k+s) to cell (j,k) with respect to array configuration c*, as follows,

1) It is 'I', if cell (j+r,k+s) contains $\rho_{u,v}$ or $\sigma_{u,v}$ in array configuration c, for $|u|+|v| \geqslant 1$.

2) It is Z_1, if cell (j+r,k+s) contains $\rho_{u,v}Z_1\ldots Z_t$ or $\sigma_{u,v}Z_1\ldots Z_t$ in array configuration c, for $|u|+|v|$, $t \geqslant 1$

3) If the above two cases do not hold, then it is the contents of cell (j+r,k+s) in array configuration c

(Notice that a cell may be in both active and passive transmission mode or in both active and passive acceptance mode, while a wff may be both an active and passive message between the same two cells.)

Now, suppose that the contents of cell (j,k) in array configuration c is Z, and that c is the array configuration of G# at time $t = t_0$. Suppose, also, that at this time, cell (j,k) is receiving the external input X. We now specify the contents of cell (j,k) in array configuration c', the successor array configuration of c; that is, the array configuration of G# at time $t = t_0 + 1$.

1) Suppose Z is of the form '$\rho_{r,s}Y_1 \ldots Y_m$', for $|r|+|s| \geqslant 1$, $m > 1$

a) If cell (j,k) is in a passive acceptance mode from only one cell with respect to array configuration c, and this cell is (u,v), then the contents of cell (j,k) in array configuration c' is either the normal form of $X(U(Y_2 \ldots Y_m))$, where U is the active message from cell (u,v) to cell (j,k) with respect to array configuration c, if such a normal form exists, or is 'I' otherwise

b) If cell (j,k) is either not in a passive acceptance mode from any cell with respect to array configuration c, or is in a passive acceptance mode from more than one cell with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is either the normal form of $X(Y_2 \ldots Y_m)$, if such a normal form exists, or is 'I' otherwise

2) Suppose Z is of the form '$\rho_{r,s}Y_1$', for $|r|+|s| \geqslant 1$

a) If cell (j,k) is in a passive acceptance mode from only one cell with respect to array configuration c, and this cell is (u,v), then the contents of cell (j,k) in array configuration c' is either the normal form of XU, where U is the active message from cell (u,v) to cell (j,k) with respect to array configuration c, if such a normal form exists, or is 'I' otherwise

b) If cell (j,k) is either not in a passive acceptance mode from any cell with respect to array configuration c, or is in a passive acceptance mode from more than one cell with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is X

3) Suppose Z is of the form '$\sigma_{r,s}Y_1 \ldots Y_m$, for $|r|+|s|$, $m \geqslant 1$

a) If cell (j,k) is in a passive acceptance mode from any cell, different from (j+r,k+s), with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is either the normal form of $X(Y_1 \ldots Y_m)$, if such a normal form exists, or is 'I' otherwise

b) If cell (j,k) is not in a passive acceptance mode from any cell, different from (j+r,k+s), with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is either the normal form of $X(U(Y_1 \ldots Y_m))$, where U is the passive message from cell (j+r,k+s) to cell (j,k) with respect to array configuration c, if such a normal form exists, or is 'I' otherwise

4) Suppose Z is of the form '$\sigma_{r,s}$', for $|r|+|s| \geqslant 1$

a) If cell (j,k) is in a passive acceptance mode from any cell, different from (j+r,k+s), with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is X

b) If cell (j,k) is not in a passive acceptance mode from any cell, different from (j+r,k+s), with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is either the normal form of XU, where U is the passive message from cell (j+r,k+s) to cell (j,k) with respect to array configuration c, if such a normal form exists, or is 'I' otherwise

5) Suppose none of the above cases hold

a) If cell (j,k) is in a passive acceptance mode from only one cell with respect to array configuration c, and this cell is (r,s), then the contents of cell (j,k) in array configuration c' is either the normal form of $X(UZ)$, where U is the active message from cell (r,s) to cell (j,k) with respect to array configuration c, if such a normal form exists, or is 'I' otherwise

b) If cell (j,k) is either not in a passive acceptance mode from any cell with respect to array configuration c, or is in a passive acceptance mode from more than one cell with respect to array configuration c, then the contents of cell (j,k) in array configuration c' is either the normal form of XZ, if such a normal form exists, or is 'I' otherwise

An Outline of the Proof of the Main Theorem in Two Dimensions

Suppose G = <S,E,2,α,β> is a 2-dimensional g.t.a. which is of countable state size, f.v.n.s., f.v.t., of finite neighborhood bound δ, c.u.n.s. c.u.t., n.d., t.d. and t.e. We now show that G# uniformly simulates G in δ+2 times real time.

Since S is countable, there is an injective map θ from S onto a recursive subset of N = {0,1,2,...}, which is onto if S is denumerable. In the simulation <Δ,Γ>, we define Δ:CON → CON# and Γ:INP → $\mathrm{INP\#}^{\delta+2}$ as follows,

1) If c ε CON, then, for i,j ⩾ 0, Δ(c)((2i+1,j))=C$\mathrm{I}n$, where n is the wff corresponding to the integer n = θ(c((i,j))), and Δ(c)((2i,j)) = $Z_{i,j}$, which we will presently define. $Z_{i,j}$ will in some sense represent the neighborhood index and the local state transition function which applies at cell (i,j) of c. (Note that since G is c.u.n.s., c.u.t., n.d., and t.d., we must have that α and β depend just on position.)

2) If γ ε INP, m ε δ+2, and i,j ⩾ 0, then, Γ(γ)(m)((i,j)) = 'I'

Let (i*,j*) ε Z^2. Since G is of finite neighborhood bound δ, α((i*,j*),c,e), for any c ε CON and e ε E, must be equal to <(i_1,j_1),..., (i_b,j_b)>, for some b ε δ ∪ {δ}. Now, if b = δ, we put $Z_{i*,j*} = \Theta_{i*,j*}$, while if b ε δ, we put it equal to $\underbrace{\rho_{1,0}\mathrm{I}(\rho_{1,0}\mathrm{I}(\ldots(\rho_{1,0}\mathrm{I}}_{\delta\ -\ b}\Theta_{i*,j*})\ldots))$,

where $\Theta_{i*,j*}$ is a wff which we will define shortly. (The prefix of the above wff will be seen as needed in order to insure the synchronicity of the simulation.) Now, for any c ε CON and e ε E, β((i*,j*),c,e) is the same element of $\{f \mid f:S^b \to S\}$. With this function, we may associate a function $g:\theta(S)^b \to \theta(S)$, which is such that $g(n_1,\ldots,n_b) = n^* \leftrightarrow \beta((i^*,j^*),c,e)(s_1,\ldots,s_b) = s^*$, where, for $1 \leqslant i \leqslant b$, $n_i = \theta(s_i)$ and $n^* = \theta(s^*)$. Since G is t.e., we may extend g to a total recursive function h ε $\{f \mid f:N^b \to N\}$. It is known that there exists an irreducible wff h such that $h\underline{n}_1\ldots\underline{n}_b > \underline{n}^*$ if and only if $h(n_1,\ldots,n_b) = n^*$ (see Curry and Feys (9) and Church (10)). Now, let T be the wff denoted by '$\lambda UVXYZ(UV(XZ)Y)$' - for an explanation of this notation, see Church (10). We then define wffs $\Omega_1,\ldots\Omega_b$, as follows,

1) Ω_1 is $\sigma_{2i_b+1,j_b}(\mathrm{T}(\mathrm{B}\rho_{1,0}))$

2) For $1 \leqslant k \leqslant b-1$, Ω_{k+1} is $\sigma_{2i_{b-k}+1,j_{b-k}}(\mathrm{T}\Omega_k)$

We then let $\Theta_{i*,j*}$ be $\Omega_b(\mathrm{BK}(\mathrm{CI}))h(\sigma_{1,0}\Sigma)$, where Σ is an irreducible wff which has the property that $\Sigma\Phi > \Omega_b(\mathrm{BK}(\mathrm{CI}))h(\sigma_{1,0}\Sigma)$ for any wff Φ. (We may easily construct such a wff: let $\Sigma*$ be $\lambda XY(\Omega_b(\mathrm{BK}(\mathrm{CI}))h(\sigma_{1,0}(XX)))$; we then let Σ be the normal form of the wff $\Sigma*\Sigma*$.)

Now, suppose $G = <S,E,2,\alpha,\beta>$ is a 2-dimensional g.t.a. which is of countable state size, f.v.n.s., f.v.t., of finite neighborhood bound δ, c.u.n.s., c.u.t., and t.e. We now indicate how to show that G# simulates G in $\delta+3$ times real time.

Let $(i*,j*) \in Z^2$. Since G is f.v.n.s., f.v.t., c.u.n.s., and c.u.t., there is a fixed, finite number of neighborhood index, local state transition function pairs which can be applied to this cell at any time. Suppose there are t such pairs. For the i^{th} pair, $1 \leq i \leq t$, construct the wff $Z^i_{i*,j*}$ in the same manner as $Z_{i*,j*}$ was constructed above, except for the fact that the wff $\Sigma^i_{i*,j*}$, corresponding to Σ above, will be different (and will presently be specified). Now, let $Y_{i*,j*}$ be $<Z^1_{i*,j*},\ldots,Z^t_{i*,j*}>$ - the ordered t-tuple of $Z^1_{i*,j*},\ldots,Z^t_{i*,j*}$ (see Fitch (9)). Now, also from Fitch (9), we have the existence of a wff, $\mathcal{J}$, which is such that $\mathcal{J}<Z^1_{i*,j*},\ldots,Z^t_{i*,j*}>k > Z^k_{i*,j*}$, for $1 \leq k \leq t$. We now specify $\Sigma^i_{i*,j*}$, for $1 \leq i \leq t$, by letting it be such that $\Sigma^i_{i*,j*}\Phi > \Xi_{i*,j*}$, for Φ any wff and $\Xi_{i*,j*}$ the normal form of $\mathcal{J}<Z^1_{i*,j*},\ldots,Z^t_{i*,j*}>$.

In the simulation $<\Delta,\Gamma>$, we define $\Delta:\mathrm{CON} \rightarrow \mathrm{CON}\#$ and $\Gamma:\mathrm{INP} \rightarrow \mathrm{INP}\#^{\delta+3}$ as follows,

1) Δ is similar to the above except that we substitute $\Xi_{i,j}$ for $Z_{i,j}$ in the definition

2) If $\gamma \in \mathrm{INP}$ and $i,j \geq 0$, $\Gamma(\gamma)(0)((i,j)) = \mathrm{CI}\underline{k}$, where k is the index of the appropriate neighborhood index, local state transition function pair in the above ordering of all possible such pairs for cell (i,j), while for $m \in \delta+3$, $m \neq 0$, $\Gamma(\gamma)(m)((i,j)) = \text{'I'}$

Some Properties of G# (Presented without Proof)

1) G# is not transitionally effective

2) G# has no Garden-of-Eden configurations. Let $c\# \in \mathrm{CON}\#$. Let $\gamma\# \in \mathrm{INP}\#$ be such that $\gamma\#(\underline{j}) = \text{'I'}$ for all $\underline{j} \in Z^n$. Then, there is a $c \in \mathrm{CON}\#$, such that $\tau_{\gamma\#}(c) = c\#$

3) Let $c\# \in \mathrm{CON}\#$ be such that $\{\underline{j} \mid \underline{j} \in Z^n \text{ and } c\#(\underline{j}) \neq \text{'I'}\}$ is finite. Let $\gamma\# \in \mathrm{INP}\#$ be as in 2) above. Then, there is an $m \geq 0$ and a $c \in \mathrm{CON}\#$, $\{\underline{j} \mid \underline{j} \in Z^n \text{ and } c(\underline{j}) \neq \text{'I'}\}$ being a unit set, such that $\tau^m_{\gamma\#}(c) = c\#$

An Exhibition of Some Initial Configurations of G# in Two Dimensions Which Embody Various Constructions and Calculations

1) An originally 'unicellular organism' which grows alternately one cell to the right and then one cell upward.

Let the wff ζ be $\lambda XYZ(\rho_{0,1}(YXY)(XXY))$, and the wff ξ be $\lambda XYZ(\rho_{1,0}(XXY)(YXY))$. We then let Λ be the normal form of $\zeta\zeta\xi$, and the wff Π be the normal form of $\xi\zeta\xi$. Suppose we want cell (i,j) to contain our organism when it is unicellular. We then define $c\# \varepsilon$ CON# by, $c\#((i,j)) = \rho_{1,0}\Lambda\Pi$, while $c\#((i^*,j^*))$ = 'I', for $(i^*,j^*) \neq (i,j)$. Let $\gamma\# \varepsilon$ INP# be such that $\gamma\#((k,m))$ = 'I', for all $k,m \geqslant 0$. We then have that the process determined by $\{\gamma\#\}^{\omega}$ and c# embodies the aforementioned growth.

2) The evolution of a finitely marked tape being processed by a Turing machine Tu. Specifically, a representation of the tape and scanned position moves to the right, each move being another step in its evolution. If the particular Turing machine Tu doesn't eventually halt when started scanning this tape in the given position, the movement goes on indefinitely, while if Tu does eventually halt, the movement eventually stops, with a representation of the final tape and scanned position as the contents of some cell, all other cells having 'I' as contents.

Let Tu be a particular Turing machine and $\underline{t}$ a finitely marked tape for this machine. It is known that there exists wffs t, h, and f, such that t is a representation of $\underline{t}$ and the position of the scanned square, ht has either $\underline{0}$ or $\underline{1}$ as a reduction depending on whether or not Tu halts on $\underline{t}$ after scanning it in the appropriate position, respectively, and $ft > t+$, if $ht > \underline{1}$, where $t+$ is a representation of the new tape and scanning position which results. We may now define a wff Γ, which is such that Γt has either $\rho_{1,0}(K\Gamma)(\rho_{1,0}(CIt))$ or t as a reduction, depending on whether $ht > \underline{1}$ or $ht > \underline{0}$, respectively. Let $i^*,j^* \geqslant 0$. We then define $c\# \varepsilon$ CON# by, $c\#((i^*,j^*)) = \rho_{1,0}(K\Gamma)(\rho_{1,0}(CIt))$, while $c\#((i,j))$ = 'I', for $(i,j) \neq (i^*,j^*)$. Letting $\gamma\#$ be as in 1) above, we have that the process determined by $\{\gamma\#\}^{\omega}$ and c# embodies the aforementioned evolution.

REFERENCES

1) Amoroso, S., and Cooper, G., *The Garden-of-Eden Theorem for Finite Configurations*, Proceedings of the A.M.S., vol. 44 (1970), pp. 189-197
2) ———, *Tessellation Structures for Reproduction of Arbitrary Patterns*, Journal of Computer and System Sciences, vol. 5 (1971), pp. 455-464
3) Ostrand, Thomas J., *Pattern Reproduction in Tessellation Automata of Arbitrary Dimension*, Journal of Computer and System Sciences, vol. 5 (1971), pp. 623-628
4) Smith, A.R. III, *Cellular Automata Theory*, Technical Report 2, Digital Systems Laboratory, Stanford University, Stanford, California, 1969
5) ———, *Cellular Automata Complexity Trade-Offs*, Information and Control, vol. 18 (1971), pp. 466-482
6) ———, *Simple Computation-Universal Cellular Spaces*, Journal of the A.C.M., vol. 18 (1971), pp. 339-353
7) Yamada, H., and Amoroso, S., *Tessellation Automata*, Information and Control, vol. 14 (1969), pp. 299-317
8) Curry, H., and Feys, R., *Combinatory Logic, Volume I*, North Holland Publishing Company, Amsterdam, 1958
9) Fitch, Frederic B., *A System of Combinatory Logic*, Technical Report 9, Office of Naval Research, Group Psychology Branch, Contract SAR/Nonr-609(16), 1960
10) Church, A., *The Calculi of Lambda-Conversion*, Annals of Mathematical Studies, #6, Princeton University Press, Princeton, 1941

A PROCEDURE FOR DESCRIBING DISCRETE PROCESSES

MARIO R. SCHAFFNER
Massachusetts Institute of Technology, Cambridge, Massachusetts

ABSTRACT

It has been argued that for complex automata a symbolic description of the automaton itself is simpler than a symbolic description of its behavior. Experimental evidence of this conjecture has been found in using a restructurable processing machine.

The notion of a programable network is introduced and a finite state automaton suitable for modeling discrete processes is formulated. The description of the automaton provides a concise way of describing a process. The interest of this approach lies in the use of these automata as a programing language for computers that have a self-organizing capability.

1. INTRODUCTION

There is, at present, a growing number of studies on formal systems for specifying and analyzing logical activities, such as computations and processes. Undoubtedly, these studies were originally motivated by the need for a theoretical basis in programing computers; subsequently, they developed as independent branches of mathematics. On one hand, there is the general expectation that all these studies eventually will produce a comprehensive and rigorous framework for the multifaced field which is usually labeled systems or computer science. On the other hand, there is an increasing concern with the separation between these studies and the actual solution of practical problems.

There are studies that deal with a given structure, or set of rules, (which are essentially easily defined mathematically) and explore their properties and capabilities, e.g. particular automata or formal languages. However, each of these systems alone is either too restrictive, or too undefined, and lacks that flexibility so necessary in dealing with the problems of the real world.

There are studies on fascinating structures that have the capability of self-organizing, such as cellular automata. Here a broad flexibility is available, but it is achieved at the cost of a cumbersomeness that is exactly opposite to what we demand in real systems.

There is a system - the Turing machine - that does not exhibit limitations, but is unfortunately inherently tied to an inefficiency we neither can afford, nor want in any real implementation.

More recently, explicit attempts to use results from these studies for guiding the design of real systems have appeared. An incentive in this direction arises from the new possibilities due to recent technological developments. The structure of the processes and of the data are analyzed with the object of optimizing the work of the systems that will process them. Yet, these efforts appear limited to certain control aspects of the processes.

In the above context, the subject of this paper can be seen as a new approach to linking theoretical models with the solution of practical problems. In this approach, the structure of the problem is analyzed as it is in the originator's mind, and then a corresponding real machine is synthetized. Clearly, in this case, theoretical models and real problems are closely related. In fact, this study had originated from the work with an actual restructurable processing machine [1]. The descriptions of the structures that this general-purpose machine has to assume in order to execute given processes appear to be simpler than corresponding programs for obtaining these processes by conventional computers [2]. In order to formalize these results, the following procedure is given.

2. THE STARTING POINT

McCulloch and Pitts, in studying the activity of neural systems, demonstrated [3] that any activity that can be logically described can be performed by a formal network (combinational and sequential) of logical neurons. Their theorems were important in neurophysiology because they made it possible, for the first time, to produce some theoretical models for the complex activities peculiar to the human brain. This, regardless of the fact that the logical neurons considered are very poor, simplified models of the biological neurons, and the fact that those networks represent only one of the many ways (of which we still have no or little knowledge) by which neural systems process information.

Obviously, such theorems can not be irrelevant to the computer field. It was von Neumann who clearly sensed the importance of those findings for computers. His viewpoint can be summarized by the following quotation [4]:

> The McCulloch-Pitts result. . . proves that anything that can be exhaustively and unambiguously described, anything that can be completely and unambiguously put into words, is ipso-facto realizable by a suitable, finite neural network. Since the converse statement is obvious, we can therefore say that there is no difference between the possibility of describing a real or imagined mode of behavior completely and unambiguously in words, and the possibility of realizing it by a finite, formal neural network. The two concepts are coextensive. A difficulty of principle embodying any mode of behavior in such a network can exist only if we are also unable to describe that behavior completely. (1)

Furthermore, Von Neumann noted that complex activities, such as the recognition of visual analogies, are difficult to describe in ordinary language while on the contrary they are readily accomplished by the finite machine that is our brain. Then he conjectured [5] that, for very complex automata, the description of the automaton itself should be simpler than a symbolic description of its behavior, leaving an unclarified allusion that the two types of description should involve different levels of mathematical complexity.

The correspondence pointed out in quotation (1) is interesting not because we will ever construct those formal networks, but because it reveals the capability of human beings to reason in the form of abstract networks, besides the familiar forms of a spoken language and of the mathematical languages. This raises the question of which are the actual primitives in the mechanics of our thinking. Of extreme interest on this subject are the studies made independently on human thoughts and expressions by philosophers, psychologists and linguists [6],[7],[8],[9],[10],[11],[12]. These studies are now using more and more of the same mathematical methods by which we can expect more and more connections between the different sciences. Unfortunately, these studies are still quite far from being at the stage where we can derive practical guidance for our subject.

On the other hand, we observe in everyday life that when a complex process has to be described, or analyzed, people instinctively draw abstract schemes of the process on a sheet of paper or on a blackboard. We see, as a basic resource of human beings, the expression of what they think in the form of abstract structures, materialized by means of sketches. Whether this habit derives from the continuous use of the sophisticated vision apparatus we possess in order to interact with the environment, or from the inner structure at some level of our thinking is irrelevant to our application.

Suggested by the above observations of everyday life, and supported by the rigorous correspondence found between mathematical functions and abstract machines (e.g. Turing machines), and between languages and abstract machines (e.g. correspondence between formal languages and automata), we imply in this paper the existence of an innate facility of human beings for thinking of processes as abstract machines. Then we will search for a rigorous procedure for exploiting this hypothesized facility,i.e., a rigorous formulation of those abstract machines. Finally, we will use this rigorous form as a language for communicating with the man-made machines that we call computers.

3. NOTIONS AND DEFINITIONS

Network. The McCulloch-Pitts formal networks were composed of logical devices intended as axiomatic simplified models of the biological neurons, but, as is well known, many other logical devices can be used for the same purpose (e.g. the now-familiar NAND devices in conjunction with delay units). Moreover, by the proper assembling of these devices, other elements can be produced that exhibit higher-level functions, more similar to those we use in thinking of processes.

In this paper we will call logical network, or simply network, any logical network that, regardless of the devices used and the level of the functions considered, corresponds to a logical activity in the sense pointed out in quotation (1). If F_j stands for a logical description of an activity j, and N_j for a logical network performing that activity, we can thus consider the mapping:

$$F_j \rightarrow N_j \tag{2}$$

Programable Network. Let us now think of a very large (finite) logical network composed of operating and storage elements. For simplicity, and without loss of generality, we assume binary elements and signals. Such a network can be regarded as a giant, unmanageable, finite-state machine with a very large number of states and input signals. The state diagram of such a machine is undescribable, we call it the total state diagram. Let us divide the input signals into two categories which we call process inputs and p inputs. For each set of values of the p inputs, a particular "component" (using a Minsky [13] terminology) of the total state diagram is selected, while all the rest disappears. Such a component can be regarded as a specific computation.

If there are m of such p inputs, we can look at the ensemble of the values at these inputs as a word W of m binary digits. All the possible patterns for these digits form a set of 2^m words. Let us call L the subset of these words that correspond to meaningful logical networks. We can thus consider, in the subset L, the mapping

$$N_j \rightarrow W_j \tag{3}$$

In the sequel we will call programable network (PN) a logical network as described, where words W_j can implement specific networks N_j. We consider the subset L as the language of that programable network, and the mapping

$$F_j \rightarrow W_j \tag{4}$$

as the semantics of that language.

Finite-State Machines. The usual formal definition of a finite-state machine (fsm) consists of a quintuplet [B, Z, Q, F, T] where B and Z are the finite input and output alphabet respectively, Q a finite set of states, F: $Q \times B \rightarrow Z$ and T: $Q \times B \rightarrow Q$ the characterizing functions of the fsm. Sometimes, when the fsm is used simply as an acceptor, it is defined by a quintuplet [B, Q, Q', F, T] where Q' is the set of final states. Because of the use we are going to make of the fsm, we formulate them slightly differently.

(i) We divide the total states Q into partitions s, which we call the "states" of the machine, and we look at the substates in each partition as values of internal variables that are stored inside the machine. We indicate as a vector $X = x_1, x_2, \ldots x_K$ the ensembles of these internal variables.

(ii) In the variety of applications in which we will use the fsm modeling, functions F and T will typically have a very different form for each value of the "state s"; therefore, for ease of expression, we write F and T as independent functions for each value of s. In this way the explicit dependency of F and T on s is removed.

(iii) With regard to the output, in our fsm we consider two cases:
(a) Some of the internal variables X are used also as output data. The machine might never stop, and there is an endless production of output.
(b) One of the states will be called the halt state; when the machine transfers to that state, it stops, and the present internal variables are considered as the output data. Obviously, a machine can have both of these output features.

Throughout this paper, the input, output, and internal-variable alphabets are considered to be the same and to consist of the numbers from 0 to 2^r; therefore, the definition of the alphabets will not be restated hereafter.

Thus, our fsm is defined by a finite set of doublets

$$[F_j, T_j] \qquad j = 0,1,2,\ldots.k \tag{5}$$

where j is the state s, and F_j, T_j are the expressions of the characterizing functions for the state j. At each discrete time i the machine produces new values for the internal variables X, and the next state s(i+1), in accordance with the expressions

$$\begin{aligned} X(i+1) &= F_{s(i)}\,[U(i),X(i)] \\ s(i+1) &= T_{s(i)}\,[U(i),X(i+1)] \end{aligned} \tag{6}$$

where U is a set of symbols constituting the present process-input, and s(i) the present state. At the starting state, X is supposed to have 0 value; then, the expressions of functions F and T, implicitly, introduce the internal variables of the process. A finite-state machine formulated in this way will be denoted as an FSM hereafter.

4. FSM AUTOMATON

A programable network PN, as defined in Section 3, is embedded with a set X of internal variables x_v, Fig. 1. By the term embedded it is meant that the variables are expressed in some digital form and are stored in some elements of the network where they can be read and written by some input and output of the network. The p inputs of PN read digital words W in a storage P. Some inputs of PN are connected to the input lines ξ, and some outputs of PN are connected to the output lines ζ.

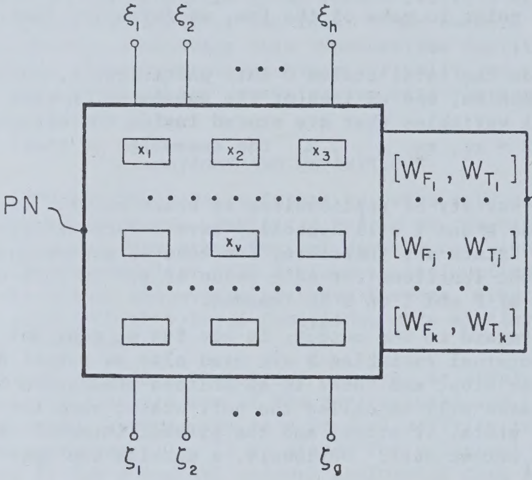


Fig. 1 - The FSM automaton

Such a structure is an FSM, as defined in Section 3, if:

(i) Doublets (5) describing an FSM are expressed in the language L of PN and stored in P.

(ii) At each discrete time i, a set of input data U(i) is presented at the input lines ξ.

(iii) At each discrete time i, two networks are implemented in PN, a first one in response to a word W_F for determining a new value of X, and a second one in response to a word W_T for determining a doublet (next state) from the set in P.

(iv) Between times i, the previously determined doublet is connected to the p inputs of PN, it being understood that for i = 1 the doublet is the one that has been indicated as the starting state.

(v) The halt state implements a network that connects a subset of X to the output lines ζ.

This automaton, which can implement a variety of FSMs, in response to sets of doublets stored in it, will be called an FSM automaton.

If a particular FSM, which we will call FSM_A, described by k states, is the model of an activity A (in other words, FSM_A corresponds to activity A in the sense discussed in Section 2), we can obtain this activity from an

FSM automaton if the language L of the automaton permits description of the k states of FSM_A by k doublets (5). In general, a given FSM automaton will be capable of producing an activity A, if we are able to describe in the language of that automaton an FSM of $k' \geqslant k$ states equivalent to FSM_A of k states, where FSM_A is the model that the user conceived for the activity A. Two FSMs are said equivalent here when they produce the same output result.

Note that here finite-state machines (our FSMs) are not used for representing a language (which eventually will be used for describing the desired activity). Neither are they used for modeling the activity of a circuit (which eventually will simulate our process or a part of it). The FSMs are used for modeling the desired activity A; in other words, for defining the activity A.

The FSMs themselves are described by means of a language appropriate for their description,which is always the same regardless of the activity of which an FSM is the model. This language should have the capability of describing the elements of which the FSMs are constituted: the states and their components F and T. The syntactic structure of this language can be of an elementary simplicity: the entire description of an FSM is composed of a finite number of doublets; the doublets are listed in the order of the label of the state they represent; and in each doublet the first word refers to the function F, and the second to the function T. The word morphology of F and T, on the other hand, is more complex.

A programable network demands that all the information for its definition be provided simultaneously, by means of words W. Successful experiments have been made [14] in using several features of spoken languages - such as agglutination, conjugation, prefixes and suffixes - for conveying a large amount of information by means of words that are constructed by the user for each occasion.

In the actual implementation of programs, [2], [18], it appears that the description of a complex activity by means of a language describing its FSM model is simpler than the descriptions by means of languages describing the activity itself, as is the case in conventional programing languages.

5. CPL AUTOMATON

Turing showed the power of an infinite tape added to a finite-state machine. But, if we are interested in realizable machines, we can not use infinity. An alternative solution is here considered, as shown in Fig. 2, [1]; the FSM automaton of Section 4 is provided with a Q memory — a finite First-Input-First-Output storage. All the data of an FSM, as symbolized in Fig. 1, are removed from PN and P at each transition, and packed in a block, called a page, into the Q memory, along with the indication of the next state and of the allocation of the x_v in PN. Thus, when a page returns into the black box, all the x assume the locations they had before, thus the process of that FSM continues as if it were never interrupted.

Several FSMs are circulating through the FSM automaton and the Q memory, either independently or concurrently, exchanging data through the page storage H which is part of the programable network. The connections with the environment allow the pages to use additional storage, outside. Because the peculiar characteristics of this machine are the organization of each FSM in the form of a page, the automatic circulation of the pages, and the looseness (programability) of the structure, it has been named the Circulating Page Loose (CPL) automaton.

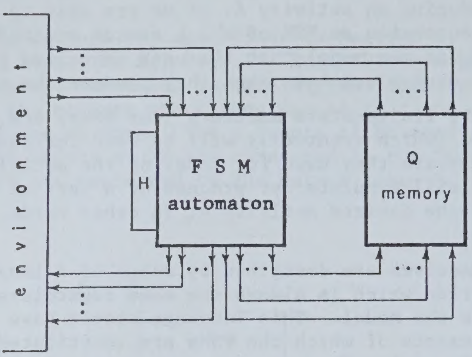


Fig. 2 - The CPL automaton

Each FSM is active only when it is in the PN, thus for an observer dealing with an FSM, all the data stay in the array shown in Fig. 1. Therefore, the description of the FSMS for the CPL automaton is similar to that for the FSM automaton.

Discussion

Note that the states have a different nature depending upon where we look within the automaton. They are strings of symbols (the doublets W_F, W_T) in the storage P; they are networks in the PN; and they are computations if we think of the pages. We see that the notion of programable networks allows us, formally, through the mapping (1) and (2), to go directly from the specification of a function to its execution, without being involved in any sort of simulation as, for example, with universal Turing machines.

The mapping (3) is implemented directly by the automaton, without intermediate activities such as compilation. Obviously, some activity in place of the compilation is performed by the user; but, because the starting points are the abstract machines in the user's mind, we argue [14] that these activities are natural for the user. Functions can be easily expressed in the form of symbolic networks simply by thinking of the inner structure of the function. The words W can be derived from the symbolic networks by the user, with an activity in which he is well trained, if the language L has been structured according to the spoken languages.

There is the question of the finiteness of the programable network, but we can extend it, virtually, by using a plurality of pages. There is the finiteness of the Q memory, but we can use the same argument as applied in relation to certain Turing machines [15]: we start with a finite tape, say closed in a loop, and then we add further pieces as required. It is precisely the strong connection with the environment that allows us to simulate additional pieces of PN and Q memory.

If we look at this automaton as a model for a real processing machine, we note two interesting facts: (1) Because a network corresponds to a large portion of computation, the description of an entire process becomes concise and easily understood. (2) Because the data relevant to each moment of a computation are in the network, many activities for data management are eliminated; e.g., the address manipulations of conventional computers.

6. THE CPL SYSTEM

An actual machine implementing the philosophy of the CPL automaton has been constructed [1]. To facilitate the communication with the environment, two units are added, Fig. 3: an assembler, which provides for the acquisition of new data; and a packer, which provides for the routing of the data produced. Accordingly, the programs consist of quadruplets [I,F,T,R], where F and T have the same meaning as in (5), I is a prescription of new input for the assembler, and R is a prescription of routing for the packer. This type of real machine has been named the CPL system.

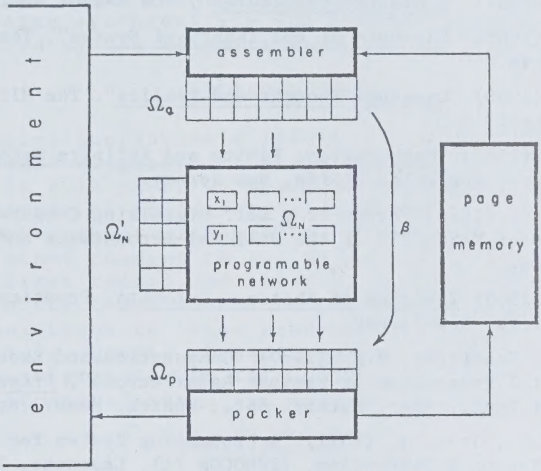


Fig. 3 - The CPL system

The results obtained in using this system for processing in real-time radar-signals in research projects at the Smithsonian Astrophysical Observatory and at M.I.T. [2],[16],[17],[18],[19], and the apparent possibility of broader applications, [14],[20], have motivated the study of the approach outlined in this paper.

REFERENCES

1. Schaffner, M.R. (1971) "A Computer Modeled after an Automaton", Proc. Symp. Computers and Automata, MRI Symp. Vol. XXI, Polytechnic Institute of Brooklyn, Brooklyn, New York, pp. 635-650.
2. Schaffner, M.R. (1972) "Computers Formed by the Problems, rather than Problems Deformed by the Computers", Digest 6th IEEE Int. Computer Conf., San Francisco, Cal., pp. 259-264.
3. McCulloch, W.S., Pitts, W. (1943) "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bull. Math. Biophys., 5, pp. 115-133.
4. Von Neumann, J. (1948) "The General and Logical Theory of Automata" Hixon Symposium, Pasadena, Cal., reprinted in A.H. Taub, "John von Neumann Collected Works", Vol. V, pp. 288-328, Pergamon Press, 1961.

5. Von Neumann, J. (1966) "Theory of Self-Reproducing Automata", edited and completed by A.W. Burks, Un. of Illinois Press, Urbana, Ill.

6. Mandler, J.M., Mandler, G. (1964) "Thinking, from Association to Gestalt", John Wiley & Sons, Inc.

7. Miller, G.A. (1964) "Mathematics and Psychology", John Wiley & Sons, Inc.

8. Piaget, J. (1950) "The Psychology of Intelligence", Routled & Kegan Paul Ltd., London.

9. Blumenthal, A.L. (1970) "Language and Psychology, Historical Aspects of Psycholinguistics", John Wiley & Sons, Inc.

10. Chomsky, N. (1957) "Syntactic Structures", The Hague: Mouton & Co.

11. Chomsky, N. (1965) "Aspects of the Theory of Syntax", The MIT Press, Cambridge, Mass.

12. Whorf, B.L. (1956) "Language Thought and Reality", The MIT Press, Cambridge, Mass.

13. Minsky, M.L. (1967) "Computation: Finite and Infinite Machines". Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

14. Shcaffner, M.R. (1972) "Study of a Self-Organizing Computer", Final Report, contract NASW-2276 of the National Aeronatuics and Space Administration.

15. Arbib, M.A. (1969) Theories of Abstract Automata. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

16. Austin, P.M., Schaffner, M.R. (1970) "Computationsand Experiments Relevant to Digital Processing of Weather Radar Achoes", Prepr. 14th Weather Radar Conf., Amer. Meteor. Soc., Boston, Mass., pp. 375-380.

17. Schaffner, M.R., Joss, J. (1971) "A Processing System for Weather Radar Data", IEEE Region 8 Convention, (EUROCON 71), Lausanne, Switzerland.

18. Schaffner, M.R. (1972) "On the Data Processing for Weather Radar" Prepr. 15th Weather Radar Conf., Amer. Meteor. Soc., pp. 368-373.

19. Schaffner, M.R. (1972) "Movement and Evolution of Echo Pattern from Real-time Processing of Radar Signals", Prepr. 15th Weather Radar Conf., Amer. Meteor. Soc., pp. 374-378.

20. Cappetti, I., and Schaffner, M.R. (1972) "Structure of a Communication Network and Its Control Computers", Proc. Symp. Computer-Communications Networks and Teletraffic, M.R.I., Vol. XXII, Polytechnic Institute of Brooklyn, Brooklyn, New York.

EXPERIMENTS ON SEQUENTIAL MACHINES

GERHARD F. BECKHOFF
Department of Computer Science
The University of Western Ontario
London 72, Canada

ABSTRACT

With every automaton are defined four associated automata, one of which in particular is used to solve in a straightforward and (conceptually and technically) simple manner the problems of state minimization, synchronization and the initial state as well as final state identification of finite sequential machines. The diagnosing, synchronizing and homing sequences are recursively determined by use of partitions only (instead of set systems).

INTRODUCTION

Recently, considerable interest has been devoted to the study of fault diagnosis of sequential machines and computer hardware. In this paper we shall be concerned with the experimental analysis of finite-state machines. The Mealy machine has been used throughout although all results can be applied equally well with minor changes to the Moore model. The problems treated here were first raised and attacked by E.F. Moore and have subsequently attracted a number of other researchers. Some of the proposed solutions to these problems have appeared in standard textbooks (e.g. Hennie, Kohavi). Contrary to the existing ones, the solutions proposed in this paper are recursive in nature, therefore straightforward and easily programmable.

BASIC DEFINITIONS

A (finite) sequential machine is a quintuple $M = (Q,I,K,\delta,\lambda)$, where Q,I,K are non-empty (finite) sets, the state set, the input alphabet and the output alphabet, respectively, $\delta:Q\times I\to Q$ is the transition function and $\lambda:Q\times I\to K$ (Mealy model) is the output function. The functions δ and λ are extended to $\delta:Q\times I^*\to Q$, $\lambda:Q\times I^*\to K^*$ and $\mu:Q\times I^+\to K$ in the following recursive way:

$(\forall q\varepsilon Q)(\forall i\varepsilon I)(\forall x\varepsilon I^*)\ \delta(q,\Lambda) = q,\ \delta(q,xi) = \delta(\delta(q,x),i)$
$\lambda(q,\Lambda) = \Lambda,\ \lambda(q,xi) = \lambda(q,x)\ \lambda(\delta(q,x),i)$
$\mu(q,xi) = \lambda(\delta(q,x),i)$.

The triple $A = (Q,I,\delta)$ is called an automaton. With every automaton A (or sequential machine M) we associate four other automata

(i) $\mathcal{P}(A) = (\mathcal{P}(Q),I,\mathfrak{f})$
(ii) $\mathcal{P}^+(A) = (\mathcal{P}(Q),I,\mathfrak{f}^+)$
(iii) $P(A) = (P(Q),I,f)$
(iv) $P^+(A) = (P(Q),I,f^+)$

where $\mathcal{P}(Q)$ and $P(Q)$ are the set (Boolean lattice) of all subsets of Q and the set (summetric partition lattice) of all partitions*) on Q, respectively. The transition functions of the associated automata are defined as follows:
for $i\varepsilon I$, $q,q'\varepsilon Q$, $Q'\varepsilon\mathcal{P}(Q)$, $\pi\varepsilon P(Q)$

---

*) A partition and its corresponding equivalence relation will be used interchangeably and will be denoted by the same symbol.

(i) $\mathfrak{f}(Q',i) = \{q\varepsilon Q \mid (\exists q'\varepsilon Q')\delta(q',i) = q\}$
(ii) $\mathfrak{f}^+(Q',i) = \{q\varepsilon Q \mid (\exists q'\varepsilon Q')q' = \delta(q,i)\}$
(iii) $q\pi q' \Leftrightarrow \delta(q,i) f(\pi,i)\ \delta(q',i)$
(iv) $\delta(q,i)\ \pi\delta(q',i) \Leftrightarrow q\ f^+(\pi,i)q'$

These functions are extended in the following recursive way:

$$(\forall Q'\varepsilon \mathcal{P}(Q))(\forall \pi\varepsilon P(Q))\ (\forall x\varepsilon I^*)(\forall i\varepsilon I)$$

(i) $\mathfrak{f}(Q',\Lambda) = Q'$, $\mathfrak{f}(Q',xi) = \mathfrak{f}(\mathfrak{f}(Q',x),i)$
(ii) $\mathfrak{f}^+(Q',\Lambda) = Q'$, $\mathfrak{f}^+(Q',ix) = \mathfrak{f}^+(\mathfrak{f}^+(Q',x),i)$
(iii) $f^+(\pi,\Lambda) = \pi$, $f(\pi,xi) = f(f(\pi,x),i)$
(iv) $f^+(\pi,\Lambda) = \pi$, $f^+(\pi,ix) = f^+(f^+(\pi,x),i)$

Note that the functions $\mathfrak{f}^+$ and f^+ are extended in "reverse" order; to obtain formulae similar to $\mathfrak{f}$ and f take the mirror image $\bar{x}$ of x:

$$\mathfrak{f}^+(Q',\overline{xi_n}) = \mathfrak{f}^+(\mathfrak{f}^+(Q',\bar{x}),i_n)$$
$$f^+(\pi,\overline{xi_n}) = f^+(f^+(\pi,\bar{x}),i_n)$$

The functions $\mathfrak{f}^+$ and f^+ can readily be determined from the state predecessor table, also called inverse state table, listing the sets $\delta^{-1}(q,i)$ of states which under the action of the input i go into the state q. The functions $\mathfrak{f}$ and f are residuated mappings on the lattice $(\mathcal{P}(Q),\cap,\cup,\Phi,Q)$ and $(P(Q),\wedge,\vee,0,u)$, respectively, and the functions $\mathfrak{f}^+$ and f^+ are their (unique) residuals. Therefore, $\bigcup_{i\varepsilon I} \mathfrak{f}(Q',i)$ and $\bigvee_{i\varepsilon I} f(\pi,i)$ are also residuated and $\bigcap_{i\varepsilon I} \mathfrak{f}^+(Q',i)$ and $\bigwedge_{i\varepsilon I} f^+(\pi,i)$ are their corresponding residuals. Moreover, for any finite index set J the functions $\mathfrak{f}$ and f are join-preserving and the functions $\mathfrak{f}^+$ and f^+ are meet-preserving. Thus:

$$\bigcup_{j\varepsilon J} \mathfrak{f}(Q_j,i) = \mathfrak{f}(\bigcup_{j\varepsilon J} Q_j,i)$$
$$\bigvee_{j\varepsilon J} f(\pi_j,i) = f(\bigvee_{j\varepsilon J} \pi_j,i)$$

and dually,

$$\bigcap_{j\varepsilon J} \mathfrak{f}^+(Q_j,i) = \mathfrak{f}^+(\bigcap_{j\varepsilon J} Q_j,i)$$
$$\bigwedge_{j\varepsilon J} f^+(\pi_j,i) = f^+(\bigwedge_{j\varepsilon J} \pi_j,i)$$

A BASIC THEOREM

For $x\varepsilon I^*$ let λ_x be the equivalence relation on Q defined by

$$q\lambda_x q' \Leftrightarrow \lambda(q,x) = \lambda(q',x).$$

This equivalence relation can be obtained recursively.

Proposition 1: Let $M = (Q,I,K,\delta,\lambda)$ be a sequential machine and $P^+(M) = (P(Q),I,f^+)$ be its associated automaton (of type iv) For $x = i_1 i_2 \ldots i_n \varepsilon I^+$ let $\overline{x}$ denote the mirror image of x, $\overline{x} = i_n,\ldots,i_2,i_1$. Then

$$\lambda_{\overline{i_1,\ldots,i_n}} = \lambda_{i_n} \wedge f^+(\lambda_{\overline{i_1,\ldots,i_{n-1}}}, i_n)$$

with $\lambda_\Lambda = u$, the universal equivalence relation.

Proof: $q \; \lambda_{\overline{i_1,\ldots,i_n}} \; q' \Leftrightarrow q \; \lambda_{i_n,\ldots,i_1} \; q' \Leftrightarrow$

$$\lambda(q,i_n,\ldots,i_1) = \lambda(q',i_n,\ldots,i_1) \Leftrightarrow$$

$$\lambda(q,i_n) \; \lambda(\delta(q,i_n),i_{n-1},\ldots,i_1) = \lambda(q',i_n) \; \lambda(\delta(q',i_n),i_{n-1}\ldots i_1)$$

$$\Leftrightarrow \lambda(q,i_n) = \lambda(q',i_n) \text{ and } \lambda(\delta(q,i_n),i_{n-1},\ldots,i_1)$$

$$= \lambda(\delta(q_1',i_n),i_{n-1},\ldots,i_1)$$

$$\Leftrightarrow q \; \lambda_{i_n} \; q' \text{ and } \delta(q,i_n) \lambda_{i_{n-1},\ldots,i_1} \; \delta(q',i_n)$$

$$\Leftrightarrow q \; \lambda_{i_n} \; q' \text{ and } q \; f^+(\lambda_{i_{n-1},\ldots,i_1},i_n) q'$$

$$\Leftrightarrow q(\lambda_{i_n} \wedge f^+(\lambda_{i_{n-1},\ldots,i_1},i_n)) \; q'$$

$$\Leftrightarrow q(\lambda_{i_n} \wedge f^+(\lambda_{\overline{i_1,\ldots,i_{n-1}}},i_n)) \; q'$$

This theorem is used in the proof of almost all the subsequent propositions.

STATE MINIMIZATION

Let $M = (Q,I,K,\delta,\lambda)$ be a sequential machine. For $k \geq 1$, states $q,q' \varepsilon Q$ are said to be k-equivalent.

$$q \; \sigma_k \; q' \Leftrightarrow \lambda(q,x) = \lambda(q',x) \quad (\forall \; x \varepsilon I^k).$$

The states q and q' are called (state-) equivalent if they are k -equivalent for all k, or equivalently

$$q \, \sigma \, q' \Leftrightarrow \lambda(q,x) = \lambda(q',x) \quad (\forall \; x \varepsilon I^*)$$

σ is the state equivalence relation on M.

Proposition 2: Let $M = (Q, I, K, \delta, \lambda)$ be a sequential machine. Two states q and q' of M are k-equivalent ($k \geq 1$) if and only if

$$\sigma_k = \bigwedge_{x \varepsilon I^k} \lambda_x = \sigma_1 \wedge \bigwedge_{i \varepsilon I} f^+(\sigma_{k-1}, i)$$

with $\sigma_0 = u$ and $\sigma_1 = \bigwedge_{i \varepsilon I} \lambda_i$

Proof: $q \sigma_k q' \Leftrightarrow \lambda(q,x) = \lambda(q',x) \quad (\forall x \varepsilon I^k)$

$$\Leftrightarrow q \lambda_x q' \quad (\forall x \varepsilon I^k) \Leftrightarrow q \bigwedge_{x \varepsilon I^k} \lambda_x q'$$

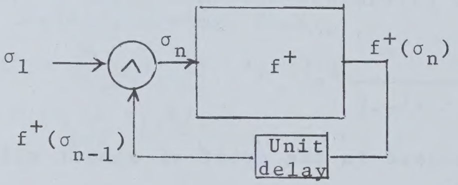
$$\Leftrightarrow q \bigwedge_{i \varepsilon I} \left\{ \lambda_i \wedge \bigwedge_{x \varepsilon I^{k-1}} f^+(\lambda_x, i) \right\} q'$$

$$\Leftrightarrow q \bigwedge_{i \varepsilon I} \lambda_i q' \text{ and } q \bigwedge_{i \varepsilon I} f^+\Big(\bigwedge_{x \varepsilon I^{k-1}} \lambda_x, i \Big) q'$$

$$\Leftrightarrow q \sigma_1 q' \text{ and } q \bigwedge_{i \varepsilon I} f^+(\sigma_{k-1}, i) q'$$

$$\Leftrightarrow q \sigma_1 \wedge \bigwedge_{i \varepsilon I} f^+(\sigma_{k-1}, i) q'$$

Let $\bigwedge_{i \varepsilon I} f^+(\pi, i) = f^+(\pi)$. The state minimization process can be pictured by the following feedback device



The state equivalence σ is recursively obtained and will be $\sigma = \sigma_1 \wedge f^+(\sigma_1) \wedge (f^+)^2(\sigma_1) \wedge \ldots \wedge (f^+)^N(\sigma_1)$, $N \leq |Q|$

$$= f^*(\sigma_1)$$

where

$$f^* = id \wedge f^+ \wedge (f^+)^2 \wedge \ldots \wedge (f^+)^N$$

is a residuated closure operator.

Example 1:

	0	1
A	E,0	D,1
B	F,0	D,0
C	E,0	B,1
D	F,0	B,0
E	C,0	F,1
F	B,0	C,0

Given state-output table

	0	1
A	-	-
B	F	C,D
C	E	F
D	-	A,B
E	A,C	-
F	B,D	E

Inverse state table

$$\sigma_1 = \lambda_0 \wedge \lambda_1 = \overline{ABCDEF} \wedge \overline{ACE}\ \overline{BDF} = \overline{ACE}\ \overline{BDF}$$

$$\sigma_2 = \sigma_1 \wedge f^+ (\sigma_1,0) \wedge f^+ (\sigma_1,1) =$$

$$= \overline{ACE}\ \overline{BDF} \wedge \overline{ACE}\ \overline{BDF} \wedge \overline{ABCDE}\ \overline{F} = \overline{ACE}\ \overline{BD}\ \overline{F}$$

$$\sigma_3 = \sigma_1 \wedge f^+ (\sigma_2,0) \wedge f^+(\sigma_2,1) =$$

$$= \overline{ACE}\ \overline{BDF} \wedge \overline{ACE}\ \overline{BD}\ \overline{F} \wedge \overline{ABCD}\ \overline{E}.\overline{F} = \overline{AC}\ \overline{BD}\ \overline{E}\ \overline{F}$$

$$\sigma_4 = \sigma_1 \wedge f^+ (\sigma_3,0) \wedge f^+ (\sigma_3,1) =$$

$$= \overline{ACE}\ \overline{BDF} \wedge \overline{AC}\ \overline{BD}\ \overline{E}\ \overline{F} \wedge \overline{ABCD}\ \overline{E}\ \overline{F} = \overline{AC}\ \overline{BD}\ \overline{E}\ \overline{F} = \sigma_3 = \sigma$$

INITIAL STATE IDENTIFICATION

Problem (not always solvable): Find the initial state (up to equivalence) of a specified sequential machine from its input-output behaviour.

The solution demands the search for a diagnosing sequence. An input $x \varepsilon I^*$ is said to be a diagnosing sequence for a sequential machine over the input alphabet I if the output sequence produced by M in response to x is different for each non-equivalent state.

Proposition 3: Let $M = (Q,I,K,\delta,\lambda)$ be a sequential machine, σ be the state equivalence relation on M, and $x \varepsilon I^+$ be an input sequence to M. Then the following conditions are equivalent:

(i) x is a diagnosing sequence for M.

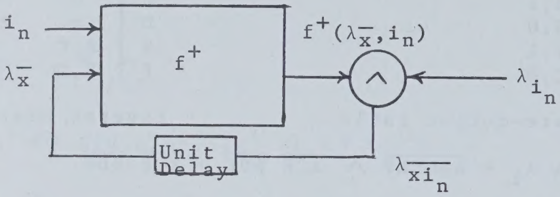
(ii) The map $\lambda(-,x) : Q/_\sigma \rightarrow K^*$ defined by $\lambda(-,x)[q]_\sigma = \lambda(q,x)$ is injective ($Q/_\sigma$ is the quotient set of Q modulo σ)

(iii) $\lambda(q,x) = \lambda(q',x) \Rightarrow q\, \sigma\, q'$

(iv) $\lambda_x \leq \sigma$

Proof: immediate.

The search process of a diagnosing sequence can be pictured by the following feedback device


i_n
$\lambda_{\overline{x}}$
f^+
$f^+(\lambda_{\overline{x}}, i_n)$
$\wedge$
λ_{i_n}
Unit Delay
$\lambda_{\overline{xi_n}}$

Let $\hat{f}^+ : P(Q) \times I \to P(Q)$ be defined by

$(\forall\, \pi \varepsilon P(Q))\ (\forall\, i \varepsilon I)\ \hat{f}^+(\pi, i) = f^+(\pi, i) \wedge \lambda_i$

Then the automaton $\hat{P}^+(M) = (P(Q), I, \hat{f}^+)$

with initial state u and set of final states *

$F = \{\pi \varepsilon P(Q) \mid \pi \leq \sigma\}$ recognizes the set of diagnosing sequences DS

$DS = \{x \varepsilon I^* \mid \hat{f}^+(u, \overline{x}) \leq \sigma\}$

Example 2:

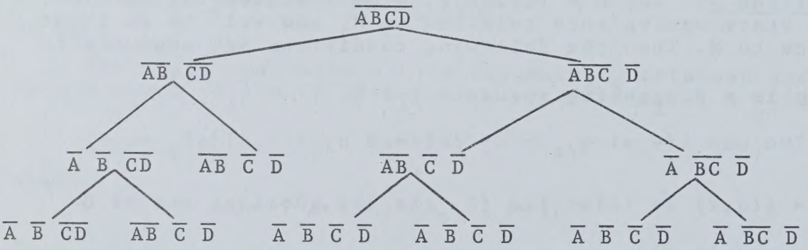
	0	1
A	C,0	D,0
B	A,0	C,0
C	D,1	B,0
D	C,1	A,1

Given state-output table

	0	1
A	B	D
B	-	C
C	A,D	B
D	C	C

Inverse state table

The diagnosing sequences can be read from the response tree of $\hat{P}^+(M)$ with initial state u and state equivalence $\sigma = 0$.


$\overline{ABCD}$
$\overline{AB}\ \overline{CD}$
$\overline{ABC}\ \overline{D}$
$\overline{A}\ \overline{B}\ \overline{CD}$
$\overline{AB}\ \overline{C}\ \overline{D}$
$\overline{AB}\ \overline{C}\ \overline{D}$
$\overline{A}\ \overline{BC}\ \overline{D}$
$\overline{A}\ \overline{B}\ \overline{CD}$
$\overline{AB}\ \overline{C}\ \overline{D}$
$\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}$
$\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}$
$\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}$
$\overline{A}\ \overline{BC}\ \overline{D}$

Diagnosing sequences of length 3:

$\overline{010} = 010$, $\overline{011} = 110$, $\overline{100} = 001$, $101 = \overline{101}$, $\overline{110} = 011$

---

*Since no partition can be less than σ, $F = \{\sigma\}$.

SYNCHRONIZATION

Problem (not always solvable): Find the final state (up to equivalence) of a specified sequential machine with unknown initial state from the knowledge of only the input sequence.

A solution exists if we can find a synchronizing sequence for M. An input $x \in I$ is called synchronizing sequence for a sequential machine $M = (Q,I,K,\delta,\lambda)$ if it forces M into a specific set of equivalent states.

Proposition 4: Let $M = (Q,I,K,\delta,\lambda)$ be a sequential machine, σ be the state equivalence relation on M and $x \in I^+$ be an input sequence to M. Then the following conditions are equivalent:

(i) The input sequence x is synchronizing for M

(ii) $(\exists\, q_x \in Q)\ (\forall\, q \in Q)\ \delta(q,x)\, \sigma\, q_x$

(iii) $(\forall\, q,q' \in Q)\ \delta(q,x)\, \sigma\, \delta(q',x)$

(iv) $f^+(\sigma,x) = u$

Proof: immediate.

The set of synchronizing sequences SS is recognized by the automaton $P^+(M) = (P(Q),i,f^+)$, having a specified initial state $\pi_0 = \sigma$ and a specified final state u. Hence

$$SS = \{x \in I^* \mid f^+(\sigma,\bar{x}) = u\}$$

The synchronizing sequences can easily be determined from the response tree of $P^+(M)$ with initial state σ.

FINAL STATE IDENTIFICATION

Problem: Find the final state (up to state equivalence) of a specified sequential machine, when its initial state is unknown, but both the input sequence and corresponding output sequence are known.

The solution demands a search for a homing sequence. An input sequence $x \in I^+$ is said to be a homing sequence for a sequential machine M over I if the response of M to x determines the final state up to state equivalence and independent of the initial state of M.

Proposition 5: Let $M = (Q,I,K,\delta,\lambda)$ be a sequential machine, σ be the state equivalence on M and $x \in I^+$ be an input sequence to M. Then the following conditions are equivalent:

(i) The input x is a homing sequence for M.

(ii) There exists a map $\Phi_x : K^* \to Q/_\sigma$ such that the following diagram is commutative:

$$\begin{tikzcd}
Q/_{\sigma} \arrow[r, "\lambda(-,x)"] \arrow[dr, "\delta(-,x)"'] & K^* \arrow[d, "\Phi_x"] \\
 & Q/_{\sigma}
\end{tikzcd}$$

where $\lambda(-,x)\colon Q/_{\sigma}\to K^*$ and $\delta(-,x)\colon Q/_{\sigma}\to Q/_{\sigma}$ are defined, respectively, by $\delta(-,x)\ [q]_{\sigma} = \lambda(q,x)$ and $\delta(-,x)[q]_{\sigma} = [\delta(q,x)]_{\sigma}$

(iii) $\lambda(q,x) = \lambda(q',x) \Rightarrow \delta(q,x)\sigma\delta(q',x)$

(iv) $\lambda_x \leq f^+(\sigma,x)$

Proof: (i) $\Leftrightarrow$ (iii) immediate.

(ii)$\Rightarrow$(iii) $\lambda(q,x) = \lambda(q',x) \Rightarrow [\delta(q,x)]_{\sigma} = \delta(-,x)\ [q]_{\sigma} =$

$$\Phi_x(\lambda(-,x)[q]_{\sigma}) = \Phi_x(\lambda(q,x)) = \Phi_x(\lambda(q',x)) =$$

$$\Phi_x\ (\lambda(-,x)[q']_{\sigma}) = \delta(-,x)[q']_{\sigma} = [\delta(q',x)]_{\sigma} \Leftrightarrow$$

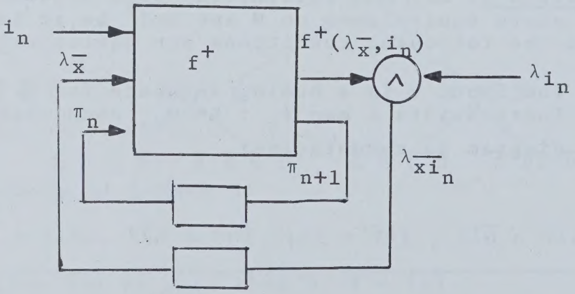
$$\delta(q,x)\ \sigma\delta(q',x)$$

(iii) $\Rightarrow$ (ii) Let $x\varepsilon I^+$ satisfy (iii). For any block $[q]_{\sigma}$ in $Q/_{\sigma}$ we define the set $W([q]_{\sigma})$ to be

$W([q]_{\sigma}) = \{z\varepsilon K^* \mid (\exists\, q'\varepsilon Q)\delta(q',x)\varepsilon[q]_{\sigma} \wedge \lambda(q',x) = z\}$ and the function $\Phi_x\colon K^*\to Q/_{\sigma}$ to be $\Phi_x(z) = [q]_{\sigma}$, if $z\varepsilon W([q]_{\sigma})$. Since $(\forall\, q,q',\varepsilon Q)\ \lambda(q',x)\varepsilon W([q]_{\sigma}) \Rightarrow \delta(q',x)\varepsilon[q]_{\sigma}, z\varepsilon W([q]_{\sigma}) \cap W([q']_{\sigma}) \Rightarrow q\sigma q'$, condition (ii) follows immediately.

(iii) $\Leftrightarrow$ (iv) $\{(\forall\, q,q'\varepsilon Q)\ \lambda(q,x) = \lambda(q',x) \Rightarrow \delta(q,x)\sigma\delta(q',x)\}$ $\Leftrightarrow \{(\forall\, q,q'\varepsilon Q)\ q\lambda_x q' \Rightarrow qf^+(\sigma,x)q'\} \Leftrightarrow \lambda_x \leq f^+(\sigma,x)$.

The homing sequences of a minimal sequential machine can be found by the following feedback device, which combines the features of the automata for recognizing diagnosing and synchronizing sequences



Let $\bar{f}^{+}: P(Q)\times P(Q)\times I \rightarrow P(Q)\times P(Q)$ be defined by

$$\bar{f}^{+}(q,q',i_n) = (\hat{f}^{+}(q,i_n), f^{+}(q',i_n))$$
$$= (f^{+}(q,i_n) \wedge \lambda_{i_n}, f^{+}(q',i_n))$$

Then the automaton $\bar{P}(M) = (P(Q)\times P(Q), I, \bar{f}^{+})$ with initial state (u,σ) and set of final states $F = \{(\pi,\pi') \varepsilon P(Q)\times P(Q) \mid \pi \leq \pi'\}$ is able to recognize the set of homing sequences HS ,

$$HS = x\varepsilon I^{*} \mid \bar{f}^{+}(u,\sigma,\bar{x}) \varepsilon F\}$$

Example 3:

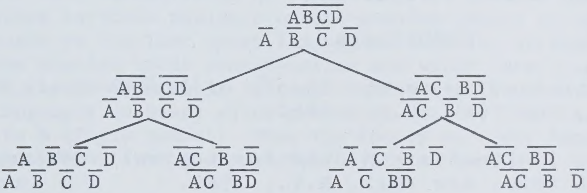
	0	1
A	B,0	C,1
B	C,0	D,0
C	D,1	C,1
D	A,1	B,0

Given state-
output table

	0	1
A	D	-
B	A	D
C	B	A,C
D	C	B

Inverse state
table

The homing sequences are easily determined from the response tree of $\bar{P}^{+}(M)$ with initial state (u,0)



Homing sequences of length 2

$\overline{00} = 00$, $\overline{01} = 10$, $\overline{10} = 01$. The sequences 00 and 01 are also diagnosing sequences.

RESTRICTED AND MULTIPLE EXPERIMENTS

While a homing sequence can always be found for a finite sequential machine, the existence of a diagnosing or synchronizing sequence is in general not guaranteed. We, therefore, consider briefly the problem of identifying the initial state of more restricted and of several machines from their input-output behaviour:

(i) A sequential machine $M = (Q,I,K,\delta,\lambda)$ with specified initial state set $Q_0 \subseteq Q$, denoted by $M(Q_0)$

(ii) A sequential machine $M = (Q, I, K, \delta, \lambda)$ with specified partition θ on Q, denoted by $M(\theta)$

(iii) A finite number k of identical copies of the same machine M, each copy being assumed to be in the same initial state (multiple experiments).

Case (i) restricts the number of possible initial states. Case (ii) requires only the identification of the θ-equivalence set containing the initial state of $M(\theta)$. In case (iii) the initial state of M is to be determined from k different input sequences. It is evident that in all three cases solutions to these problems always exist if Q_0, π and k are chosen appropriately. These solutions are

(i) $(\exists\, x \in I^*)\ \lambda_x \cap Q_0 \times Q_0 \subseteq \sigma \cap Q_0 \times Q_0$

(ii) $(\exists\, x \in I^*)\ \lambda_x \leq \pi$

Note that a solution can only exist if $\sigma \leq \pi$

(iii) $(\exists\, x \in I^*,\ i=1,\dots,k)\quad \bigwedge_{i=1}^{k} \lambda_{x_i} \leq \sigma$

Similar results are easily established for combinations of two or all cases.

ACKNOWLEDGMENT

This research has been supported by Grant A4141 of the National Research Council of Canada

REFERENCES

1. GILL, A., Introduction to the Theory of Finite-State Machines McGraw-Hill, New York, N.Y., 1962.

2. HENNIE, F.C., Finite-State Models for Logical Machines, John Wiley & Sons, New York, N.Y., 1968.

3. KOHAVI, Z., Switching and Finite Automata Theory, McGraw-Hill, New York, N.Y., 1970.

4. MOORE, E.F., Gedanken-Experiments on Sequential Machines in Automata Studies, Princeton University Press, Princeton, N.J., 1956, pp. 129-153.

5. SPIVAK, M.A., Multiple Experiments with Automata, Engineering Cybernetics, 8, (1970), 902-907.

6. STARKE, P.A., Über Experimente an Automaten, Zeitschr. f. math. Logik und Grundlagen d. Math., 13, (1967), 67-90.

ANALYTICAL REPRESENTATION OF FORMAL LANGUAGES

WILLIAM J. KELLY and R. J. LEAKE
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

INTRODUCTION

We will consider a formal language as a collection of finite strings from a given symbol alphabet. The study of such objects has been pursued from several angles: generating systems, using Chomsky's phrase-structure grammars; accepting systems, using the theory of automata; and algebraic structure, as non-commutative monoids. This paper considers a different approach, that of the analytic theory of Harmonic Analysis. This approach was suggested by studies of acceptance by finite-state-machine of sets of inteters [6] which indicate that an acceptable sequence of integers cannot have arbitrarily large gaps. This kind of characterization suggests the theory of Gap Series or Lacunary Series, studied in harmonic analysis [2,3, 7,9]. When we consider Fourier series whose non-zero coefficients have indices in such a "gap" set, we find various relations to the functions represented by the series. The idea, then, is to associate a Fourier series with the set and to investigate properties of the set by considering the functions represented by the series. The first step is to associate a series with a language. The earliest attempt to do this is apparently Chomsky and Schützenberger's work [1]. They associate a grammar with a series by considering each string as a non-commutative variable in a formal power series, and interpret integer coefficients as the degree of ambiguity of the associated grammar. The treatment, however, is purely algebraic.

This work concentrates on preserving the abelian properties of integer sets, since harmonic analysis on non-abelian groups is quite complex. For this reason we consider unary languages, that is, strings of a single symbol, which are abelian under concatenation and which have a natural map to the integers. Apart from their intrinsic interest, there is a homomorphism from n-ary languages to unary which preserves length, that is, by changing every symbol to a single symbol. Thus the theory of unary languages provides information about the characteristics of the length of strings in arbitrary languages.

As an example of this homomorphism, it can be shown that context-free unary languages are regular sets, which means that the lengths of the strings form an ultimately periodic set. When we apply the homomorphism to arbitrary languages, we conclude that the lengths of the strings in an arbitrary context-free language form an ultimately periodic set. In other words, the lengths of the strings in an arbitrary language provide a characterization of the complexity of the language.

We first consider the formalism of associating a unary language with a trigonometric series and show that in general these represent generalized functions or distributions. Then we provide a characterization of sets of language-associated distributions in terms of abstract families of distributions, and finally we look at the properties of the series associated with a non-periodic set.

UNARY LANGUAGES

Analytic Representation

Definition: A unary language L is a subset of A^* where $*$ is Kleene closure, and concatenation is the operation. The length of a string, $lg(a^n) = n$, forms a natural map to the positive integers Z^+, and thus a language L maps to a set of positive integers $E = \{n_1, n_2, \ldots\}$

Definition: Given a subset $E \subseteq Z^+$, we define the associated formal trigonometric series $f_E(n)$

$$(1) \qquad f_E(n) = \sum_{n \varepsilon E} e^{in\omega}.$$

If E is finite, $f_E(n)$ is a trigonometric polynomial. If E is not finite, it can be shown that (1) is a Fourier-Schwartz series but not a Fourier-Stietjes series, i.e., the series represents a distribution, but cannot represent a measure, much less a function. Furthermore, the distributions associated with unary languages are idempotent under convolution, that is, $\delta_1 * \delta_1 = \delta_1$, and are "one-sided" that is, all negative coefficients are zero.

Sets Closed under Operations

We now give a classification of languages which, when specialized to unary languages, is isomorphic to a classification of distributions.

Definition: [4] An abstract family of languages is a pair $(\Sigma, \mathcal{L})$ where

a) Σ is a countably infinite alphabet

b) for each $L \varepsilon \mathcal{L} \; \exists \; \Sigma_1 \subset \Sigma$, Σ_1 finite, s/t $L \subseteq \Sigma_1^*$

c) $\mathcal{L}$ is closed under union, concatenation ε-free Kleene closure, ε-free homomorphism, inverse homomorphism, and intersection with regular sets

d) $L \neq \phi$ for some $L \varepsilon \mathcal{L}$

Definition: An abstract family of monoid subsets is a pair (M, E) where

a) M is a monoid

b) $\mathcal{E}$ is a collection of subsets of M

c) $\mathcal{E}$ is closed under union, concatenation, Kleene closure, inverse homomorphism, and intersection with regular subsets.

d) $\exists E \; \varepsilon \; \mathcal{E}$ s/t $E \neq \phi$

Definition: An abstract family of distributions is a set of distributions Δ where Δ is nonempty and is closed under union, concatenation, Kleene closure, inverse homomorphism, homomorphism, and intersection with regular distributions.

The definitions of the above operations are found in Table I with the following exceptions:

i) $f: \mathcal{E} \rightarrow \mathcal{E}$ is a homomorphism if $f(e_1 \oplus e_2) = f(e_1) + f(e_2)$ and is ε-free if $f(x) = \varepsilon \Rightarrow x = \varepsilon$

ii) $g: \mathcal{E}_2 \Rightarrow 2^{\mathcal{E}_1}$ is an inverse homomorphism if $\exists f: \mathcal{E}_1 \rightarrow \mathcal{E}_2$ s/t

$g(x) = \{y \mid f(y) = x\}$ and

$g(\mathcal{E}) \triangleq \{y \mid y \; \varepsilon \; g(x) \text{ where } x \; \varepsilon \; \mathcal{E}\}$

iii) A one-sided idempotent distribution will be called regular if the subset of Z supporting its Fourier-Schwartz series is ultimately periodic.

We note that the map $\sigma: \mathcal{L} \rightarrow \Delta$ defined by $\sigma(L) = \delta(\omega) = \sum_{n \varepsilon E} e^{in\omega}$ where $E = \{lg(x) \mid x \varepsilon L\}$ is a one-to-one map onto the one-sided idempotent distributions, and commutes with the defining operations: union, concatenation, ε-free Kleene closure, ε-free homomorphism, inverse homomorphism, and intersection with regular sets. That is:

i) $\sigma(L_1 \cup L_2) = \sigma(L_1) + \sigma(L_2) - \sigma(L_1) * \sigma(L_2)$

ii) $\sigma(L_1 L_2) = \sigma(L_1) \nabla \sigma(L_2)$

iii) $\sigma(L^+) = [\sigma(L)]^+$

iv) $\sigma(f(L)) = f(\sigma(L))$

v) $\sigma(f^{-1}(L)) = f^{-1}(\sigma(L))$

vi) $\sigma(L \cap R) = \sigma(L) * \sigma(R)$

Thus the two structures are isomorphic.

LACUNARY SETS

We will now consider a class of sets which exhibit a controlled kind of non-periodicity. We want essentially to relate non-finite-state acceptors to a class of distributions, and distributions over lacunary or gap sets have the advantage that the analytic properties are related to number-theoretic properties of the integer set, and we may hope to prove some interesting theorems about such languages. We first consider Sidon sets, which are the most interesting from a harmonic analysis point of view, but which lack an arithmetic property on the integer set which is both necessary and sufficient. Finally, we look at Paley sets, which are a subset of the Sidon sets, with a more tractable arithmetic criterion. We conjecture that the largest class of non-Sidon or non-Paley languages can be shown to be contained in the context-sensitive languages.

Sidon Sets

Notation: $\underset{\wedge}{\delta}(\omega)$ is a distribution over the circle.
$\delta(n)$ is the Fourier-Schwartz transform of δ over the integers

Definition. A distribution $\delta(\omega)$ is said to be E-spectral for some $E \subseteq Z$ iff $\hat{\delta}(n) = 0 \quad n \notin E$.

Notation: T_E is the set of all E-spectral trigonometric polynomials.

Definition: E Z is called a Sidon set iff

$$\exists B_E \geq 0 \text{ s/t } ||\hat{f}||_1 \leq B_E \; ||f||_\infty \quad \forall f \in T_E$$

Structural Theorems

Notation: Given $E \subseteq Z$, enumerate as $n_1, n_2, \ldots$, where $n_i \neq n_j$ whenever $i \neq j$.

For each $n \in Z$, each $s \in Z^+$, denote by $R_s(n,E)$ the number of representations of the form

$$n = n_{k_1} \pm n_{k_2} \pm \ldots \pm n_{k_s}$$

Criterion: [3] Given $E \subseteq Z^+$, and s/t $\exists B \geq 0$ and a finite partition of E into $E_1, E_2, \ldots, E_t$, s/t $R_s(n,E_j) \leq B^s$, then E is a Sidon set.

Notation: Given $E \subseteq Z$, $N \in Z^+$, denote by $\alpha_E(N)$ the largest integer α s/t some arithmetic progression of n terms contains α elements of E: in other words, let $\alpha_E(N,a,b)$ = (number of terms of $a+b, a+2b, \ldots, a+Nb$ which fall in E) then

$$\alpha_E(N) = \sup_{a,b}[\alpha_E(N,a,b)]$$

Proposition: [3] $E \subset Z$ is a Sidon set =>

$$\alpha_E(N) \geq B_E \log N \text{ for } N \geq 3$$

It will be noted that the above necessary condition and sufficient condition are unrelated. In particular, the necessary condition is not sufficient since it can be shown [9] that ∃ non-Sidon sets E s/t for any function $\alpha(x) \to \infty$ monotonically

$$\alpha_E(N) \leq \alpha(N) \quad \forall N \in Z^+, \quad \forall N \geq 3.$$

Paley Sets

The most general class of lacunary sets for which a necessary and sufficient arithmetical condition exists seems to be the Paley sets which form a subset of the Sidon sets.

Notation: To an increasing sequence $\{n_k\}$ we associate an increasing sequence $\{\alpha_p\}$ where α_p is the number of terms of $\{n_k\}$ which satisfy $p \leq n_k \leq 2p$, for $p = 1,2,3,\ldots$

Proposition: [8] An increasing sequence $\{n_k\}$ of positive integers is a Paley sequence iff $\{\alpha_p\}$ is bounded, i.e.,

$$\lim_{p\to\infty}\{\alpha_p\} \leq M < \infty.$$

REFERENCES

1. Chomsky, N., and Schützenberger, M., "The Algebraic Theory of Context-Free Languages," in Computer Programming and Formal Systems, Braffort and Hirschberg, et. (North Holland Publishing Co., 1962).

2. Dunkl, C., and Ramirez, D., Topics in Harmonic Analysis (Appleton-Century-Crofts, 1971).

3. Edwards, R., Fourier Series V. II, (Holt, Rinehart and Winston, 1967).

4. Ginsburg, S., and Greibach, S., "Abstract Families of Languages," Memoirs of the Amer. Math. Soc., No. 89 (1969).

5. Ginsburg, S., The Mathematical Theory of Context-Free Languages (McGraw-Hill, 1966).

6. Minsky, M., and Papert, S., "Unrecognizable Sets of Numbers," J. Assn. Comp. Mach., v. 13, pp. 281-286 (1966).

7. Rudin, W., Fourier Analysis on Groups (Interscience, 1962).

8. Rudin, W., "Remarks on a Theory of Paley," J. Lond. Math. Soc., V. 32, pp. 307-311 (1957).

9. Rudin, W., "Trigonometric Series with Gaps," J. Math. and Mech., V. 9, No. 2, pp. 203-227 (1960).

TABLE I

Abstract Families Integer Sets		Abstract Families Distributions
$E_1 E_2 \varepsilon \mathcal{E} => E_1 \cup E_2 \varepsilon \mathcal{E}$	Union	$\delta_1 \delta_2 \varepsilon \Delta => \delta_1 \cup \delta_2 \varepsilon \Delta$ $\delta_1 \cup \delta_2 \triangleq \delta_1 + \delta_2 - \delta_1 * \delta_2$
$E \varepsilon \mathcal{E} \quad k \varepsilon Z =>$ $E_1 + k = \{n+k \mid n \varepsilon E\} \varepsilon \mathcal{E}$	Shift	$\delta \varepsilon \Delta \quad k \varepsilon Z =>$ $\delta_k = e^{-j\omega k} \delta \varepsilon \Delta$
$E_1, E_2 \varepsilon \mathcal{E} = E_1 E_2 \varepsilon \mathcal{E}$ $E_1 E_2 = \{n+m \mid n \varepsilon E_1 m \varepsilon E_2\}$	Concatenation	$\delta_1, \delta_2 \varepsilon \Delta => \delta_1 \nabla \delta_2 \varepsilon \Delta$ $\delta_1 \nabla \delta_2 \triangleq \lim_{N > \infty} \rho[(\sum_1^N e^{i\omega n_j})(\sum_1^N e^{i\omega m_k})]$ $\rho[\Sigma a_j e^{i\omega n_j}] \triangleq \Sigma e^{i\omega n_j}$
$E\varepsilon = E^+ \varepsilon$ $E^+ = \bigcup_1^\infty E^i$	ε-Free Kleene Closure	$\delta \varepsilon \Delta => \delta^+ \varepsilon \Delta$ $\delta^+ = \bigcup_1^\infty \delta^i$
$E \varepsilon \mathcal{E} => f_k(E) \varepsilon \mathcal{E}$ $f_k(E) = \{kn \mid n \varepsilon E, k \varepsilon Z\}$	ε-free Homomorphism	$\delta \varepsilon \Delta => f_k(\delta) \varepsilon \Delta$ $f_k(\delta(\omega)) = \delta(k\omega)$
$E \varepsilon \mathcal{E} => f^{-1}(E) \varepsilon \mathcal{E}$ $f_k^{-1}(E) = \{[\frac{n}{k}] \mid {}^{n \varepsilon E}_{[\,] \text{ nearest integer}}\}$	Inverse Homomorphism	$\delta \varepsilon \Delta => f_k^{-1}(\delta) \varepsilon \Delta$ $f_k^{-1}(\delta) = \delta([\frac{\omega}{k}])$
$E, R \varepsilon \mathcal{E}, R$ regular $=> E\ R \varepsilon \mathcal{E}$	Intersection with Regular Sets	$\delta, \delta_R \varepsilon \Delta$, δ_R regular $=> \delta * \delta_R \varepsilon \Delta$

ON THE CONTROL OF LINEAR SYSTEMS USING TWO LEVEL PERIODIC OUTPUT FEEDBACK*

RAYMOND V. WELCH and WILLIAM S. LEVINE
NASA Goddard Space Flight Center
Greenbelt, Maryland
University of Maryland
College Park, Maryland

ABSTRACT

A class of second order linear time-invariant systems which are unstable for any constant feedback but which are stabilized by a two level periodic feedback is identified. Necessary and sufficient conditions for this type of stabilization are found. In addition, the sensitivity of the closed-loop system to changes in feedback gains is investigated.

INTRODUCTION

This paper reports some preliminary results on the use of periodic piecewise constant feedback gains for the control of linear time-invariant systems. In this paper we will restrict our attention to single-input, single-output systems. As a result the open loop system can be described by a transfer function

$$G(s) = y(s)/u(s) \tag{1.1}$$

where $y(s)$ = the Laplace transform of $y(t)$, the output

$u(s)$ = the Laplace transform of $u(t)$, the input

Since we will be dealing with time-varying feedback gains, it will sometimes be more convenient to use the equivalent state-space representation

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{b}u(t) \tag{1.2}$$

$$y(t) = \underline{c}\;\underline{x}(t) \tag{1.3}$$

where $\underline{A}$, $\underline{b}$, $\underline{c}$ are a controllable and observable (minimal) realization of $G(s)$ and $\underline{A}$, $\underline{b}$ are in phse space canonical form. In most of this paper $\underline{A}$ will be a 2 x 2 matrix.

The control $u(t)$ is chosen to be an output feedback control of the form

$$u(t) = -k(t)y(t) \tag{1.4}$$

with $k(t)$ a periodic two level gain function, i. e.

---

*This research was partially supported by the National Science Foundation under grant GK-5375.

$$k(t) = \begin{cases} k_1; & nT \le t \le nT + T_1 \\ k_2; & nT + T_1 \le t \le (n+1)T \end{cases} \qquad (1.5)$$

where T_1, T, k_1 and k_2 are constants to be determined and

$n = \ldots, -2, -1, 0, 1, 2\ldots$

Thus, we will always be able to describe the closed-loop system by

$$\dot{\underline{x}}(t) = [\underline{A} - k(t)\underline{bc}]\underline{x}(t) \qquad (1.6)$$

As we have said the purpose of this paper is to investigate the use of this type of control for time-invariant linear systems. In the next section of the paper we present a motivating example due to Brockett and Lee.[1] In this example, this two level feedback is used to stabilize a linear system that cannot be stabilized by any time-invariant output feedback. In the third section of the paper it is shown that the example due to Brockett and Lee is only one of a large class of such examples. Necessary and sufficient conditions for stabilization using this type of feedback are given and proved. In addition, the extension to higher dimensions is discussed.

AN EXAMPLE

The original motivation for this study was provided by the following example, due to Brockett and Lee. Given the unstable linear time-invariant system whose open-loop transfer function is:

$$\frac{y(s)}{u(s)} = G(s) = \frac{3s-4}{s^2 - 2s + 2} \qquad (2.1)$$

If one attempts to stabilize this system by means of time-invariant output feedback

$$u(t) = -ky(t) \qquad (2.2)$$

one obtains the closed-loop system

$$G_{\text{closed-loop}}(s) = \frac{G(s)}{1+k\,G(s)} \qquad (2.3)$$

The root locus for the poles of this closed-loop system as k varies from $-\infty$ to $+\infty$ is shown in figure 1. Note that at least one pole of the closed-loop system lies in the half plane $\mathrm{Re}[s] > 0$ for all values of k. Therefore, this system cannot be stabilized by any constant output feedback gain. Now let k be time varying and satisfy

$$k(t) = \begin{cases} 0; & nT < t \le nT + T_1 \\ 1; & nT + T_1 < t \le (n+1)T \end{cases} \quad ; n = 0, 1, 2\ldots$$

where

$$T_1 = \tan^{-1}(3), \qquad T = T_1 + \pi \tag{2.5}$$

Since the resulting closed-loop system has time-varying coefficients it is useful to re-write the closed-loop system in the equivalent, controllable and observable, state space form:

$$\dot{\underline{x}}(t) = [\underline{A} - k(t)\underline{bc}]\,\underline{x}(t) \tag{2.6}$$

where

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \underline{c} = [-4 \quad 3] \tag{2.7}$$

The closed-loop system becomes

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ -2 + 4k(t) & 2 - 3k(t) \end{bmatrix} \underline{x}(t) \tag{2.8}$$

The closed-loop system (2.8) is linear with periodic coefficients. A direct application of Floquet theory (the transition matrix can be calculated analytically for this form of k(t)) proves stability.[2]

In figure 2, we illustrate a typical phase plane trajectory for the closed-loop system. Note that the straight line portions of the trajectory correspond to the eigenvectors of the closed-loop system when k(t) = 1. Note also that, when k(t) = 0, the phase plane trajectory spirals. T_1 is chosen so that any point on the unstable eigenvector is mapped into a point on the stable eigenvector in time T_1.

Finally, in fig. 3, we demonstrate that stability still holds when T_1 and $(T-T_1)$ are varied.

STABILITY

Given the second order linear time invariant differential equation,

$$\ddot{y}(t) + a_1\dot{y}(t) + a_2\, y(t) = c_2\, u(t) + c_1\dot{u}(t) \tag{3.1}$$

with transfer function

$$\frac{y(s)}{u(s)} = G(s) = \frac{c_1 s + c_2}{s^2 + a_1 s + a_2} \tag{3.2}$$

We assume throughout this section that G(s) is unstable and, as in the previous example, cannot be stabilized by any time-invariant output feedback. The question is, under what conditions can such a system be stabilized by two level periodic output feedback as described in Eqs. (1.4) and (1.5). The answer is provided by the following theorem.

Theorem: Given an unstable linear time-invariant second order system with transfer function G(s) that cannot be stabilized by time-invariant output feedback. A necessary and sufficient condition for asymptotic stabilization using two level piecewise constant periodic output feedback is that the root locus of $1 + kG(s)$ form a circle in the complex plane.

The theorem is a straightforward consequence of the following three lemmas. Note that the proof of Lemma 2 contains a construction of the stabilizing feedback.

Lemma 1 Given a system of the form (3. 1) with transfer function (3. 2) which does not have a zero and which cannot be stabilized by constant output feedback. Then there does not exist any two-level periodic output feedback gain which stabilizes the system.

Proof (We believe this result is generally known. The proof is for completeness).

Since G(s) does not have a zero $c_1 = 0$. Therefore G(s) has a minimal realization

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \underline{c} = [\, c_2 \;\; 0 \,] \tag{3. 3}$$

Thus, the closed-loop system can be written

$$\underline{\dot{x}}(t) = \begin{bmatrix} 0 & 1 \\ -a_2 \;\; -c_2 k(t) & -a_1 \end{bmatrix} \underline{x}(t) \tag{3. 4}$$

The system (3. 4) has fundamental transition matrix $\underline{\Phi}(t, t_o)$ satisfying the well known identity

$$\det \underline{\Phi}(t, t_o) = e^{\int_{t_o}^{t} \mathrm{tr}[\underline{A} - k(t)\, \underline{b}\underline{c}\,]\, dt} = e^{-a_1 (t - t_o)} \tag{3. 5}$$

But $a_1 < 0$ since G(s) cannot be stabilized by constant output feedback. Therefore, $\det \underline{\Phi}(t, t_o) \to \infty$ as $t \to \infty$. Thus, the closed-loop system is unstable.

Lemma 2 Let the system (3. 1) with the open loop transfer function G(s) of (3. 2), have a zero, $z_o > 0$ and poles λ_1 and λ_2. If λ_1 and λ_2 are complex conjugates or if λ_1 and λ_2 are real and either both greater than z_o or both less than z_o then the system (3. 1) is stabilizable with a two-level periodic output feedback gain.

Proof Under the given assumptions on G(s), the root locus of $1+kG(s)$ for $-\infty < k < \infty$ has the general form shown in figure 1 (a circle in the complex right half plane centered at the zero and straight lines on the real axis).

Thus, there exists a k_1 such the eigenvalues of $[\underline{A}-\underline{b}\,k_1\underline{c}]$ are $\alpha+j\omega$ and $\alpha-j\omega$ where:

$$\alpha,\ \omega \tag{3.6}$$

$$0<\alpha<z_o \text{ (the zero)} \tag{3.7}$$

Similarly there exists a k_2 such that the eigenvalues of $[\underline{A}-\underline{b}k_2\,\underline{c}]$ are η_1 and η_2 with the properties:

$$\eta_1,\ \eta_2 \text{ real and distinct} \tag{3.8}$$

$$\eta_1<0<\eta_2 \tag{3.9}$$

$$\tfrac{1}{2}|\eta_1|>\eta_2>\alpha \tag{3.10}$$

η_1 and η_2 have eigenvectors $\underline{e}_1$ and $\underline{e}_2$ respectively. These eigenvectors span the state space. Next, choose T_1 such that

$$e^{[\underline{A}-\underline{b}k_1\underline{c}]T_1}\,\underline{e}_2 = e^{\alpha T_1}\,\underline{e}_1 \tag{3.11}$$

This is always possible by Eqs. (3.6) and (3.7). Finally, note that, from Eqs. (3.9) and (3.10), it is always possible to choose T such that

$$(\eta_1+\eta_2)(T-T_1)+2\alpha T_1<\beta \quad (\beta \text{ any negative number}) \tag{3.12}$$

At this point it is possible to prove asymptotic stability by direct evaluation of the transition matrix. However, the following intuitive argument is more instructive. Consider an arbitrary initial vector at time T_1, $\underline{x}_o=\underline{x}(T_1)$.

$$\underline{x}_o=a_1\underline{e}_1+a_2\underline{e}_2 \qquad (a_1 \text{ and } a_2 \text{ arbitrary}) \tag{3.13}$$

$$\text{then } \underline{x}(T)=a_1e^{\eta_1(T-T_1)}\,\underline{e}_1+a_2\,e^{\eta_2(T-T_1)}\,\underline{e}_2 \tag{3.14}$$

By Eq. (3.12), T can always be chosen (in advance) such that

$$\underline{x}(T)=\epsilon\underline{e}_1+a_2e^{\eta_2(T-T_1)}\,\underline{e}_2\approx a_2e^{\eta_2(T-T_1)}\,\underline{e}_2 \tag{3.15}$$

with ϵ arbitrarily small. Then,

$$\underline{x}(T+T_1)\approx a_2e^{\alpha T_1+\eta_2(T-T_1)}\,\underline{e}_1 \tag{3.16}$$

and

$$\underline{x}(2T)\approx a_2e^{\alpha T_1+(\eta_1+\eta_2)(T-T_1)}\,\underline{e}_1 \tag{3.17}$$

$$\underline{x}(2T+T_1)\approx e^{2\alpha T_1+(\eta_1+\eta_2)(T-T_1)}\,(b_1\underline{e}_1+b_1\underline{e}_2)a_2 \tag{3.18}$$

or

$$\underline{x}(2T+T_1)\approx e^{\beta}\,(b_1\underline{e}_1+b_2\underline{e}_2)a_2$$

Since β can be chosen as negative as necessary, $\underline{x}(2T+T_1)$ is as "small" as one likes. Since the system is periodic, and cannot have a finite escape time, this proves stability. Note that, in practice, T can usually be chosen to be fairly "small".

Lemma 3: Let the system (3.1) with the open-loop transfer function G(s) of (3.2) have a zero $z_o > 0$ and real poles λ_1, λ_2 with $\lambda_1 < z_o < \lambda_2$. Then the system (3.1) cannot be stabilized with a two-level periodic output feedback gain.

Proof: The root locus for such a system always has real poles, one of which is greater than the zero and one of which is always less than the zero.

Using the identity (3.5), one can show that k_2 must be chosen such that the conditions (3.8) through (3.10) are satisfied by η_1 and η_2, the eigenvalues of $[\underline{A}-\underline{b}k_2\underline{c}]$. (If not, $\text{tr}[\underline{A}-\underline{b}k(t)\underline{c}]>0$ for all t). Associated with the unstable eigenvalue, η_2, is an eigenvector $\underline{e}_2$. We will show that a trajectory with initial condition on $\underline{e}_2$ is unstable. As long as $k(t) = k_2$, such a trajectory remains on $\underline{e}_2$ and grows. When $k(t) = k_1$ there are only two possibilities:

1) the eigenvalues of $[\underline{A}-\underline{b}k_1\underline{c}]$ are both real and positive. In this case $\underline{e}_2$ is counter-clockwise from an unstable eigenvector. Thus, a trajectory starting on $\underline{e}_2$ is bounded away from the velocity equals zero line in the phase plane.

2) one eigenvalue of $[\underline{A}-\underline{b}k_1\underline{c}]$ is positive and one is negative. In this case $\underline{e}_2$ is clockwise from the unstable eigenvector but trajectories starting on $\underline{e}_2$ tend toward the unstable eigenvector. Again, trajectories from $\underline{e}_2$ cannot reach the velocity equal zero line.

Therefore, no choice of k_1 stabilizes.

As the following example will show the possibility of stabilizing a system with periodic two-level output feedback gains is not restricted to second order systems.

Example 2: Given the third order linear time-invariant system with open loop transfer function

$$G(s) = \frac{.75s^2+3.25s-7}{s^3-2s+4}$$

The root locus for the closed loop system $1+kg(s) = 0$ for $k \in (-\infty, \infty)$ is sketched in figure 4. As seen by this root locus, at least one root of the closed loop system lies in the half plane $\text{Re}[s]>0$ for all $k \in (-\infty, \infty)$. Therefore, this system cannot be stabilized with any constant output feedback gain.

This system has a controllable and observable realization given by

(1. 2) and (1. 3) Let u(t) be defined as in (1. 4) with k(t) satisfying

$$k(t) = \begin{cases} 0; & nT < t \le nT + T_1 \\ 1; & nT + T_1 < t \le (n+1)T \end{cases} \quad ; n = 0, 1, 2, \ldots$$

and $T_1 = 1.2$, $T = 3.34$

Then the system realization can be simplified to the equation (1. 6) where

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & 2 & 0 \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{c} = [\,-7 \quad 3.25 \quad .75\,]$$

It can be shown that the resulting closed-loop system is asymptotically stable.[3]

The following example demonstrates that a different construction of the stabilizing k(t) from the one used in lemma 2 is possible.

Example 3: Given the second order differential equation

$$\ddot{y}(t) - 7\dot{y}(t) + 12y(t) = 8\dot{u}(t) - 14u(t)$$

with a controllable and observable realization given by equation (1. 3) where

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -12 & 7 \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\underline{c} = [\,-14 \quad 8\,]$$

Let k(t) be defined

$$k(t) = \begin{cases} 0 \; ; & nT < t \le nT + T_1 \\ 1 \; ; & nT + T_1 < t \le (n+1)\, T \end{cases} \quad ; n = 0, 1, 2, \ldots$$

with

$$T_1 = \ln [\, 5/4 \,]$$

$$T = T_1 + 2$$

With this choice of k(t), the eigenvalues of $[\underline{A}-\underline{b}k(t)\underline{c}]$ are 3 and 4 when $k(t) = 0$ and are 1 and -2 when $k(t) = 1$. This closed-loop system can also be shown to be asymptotically stable even though the eigenvalues are always real.[3]

SUGGESTIONS FOR FURTHER RESEARCH

It is believed that these stability results can be extended to nth order systems. The difficulty lies in relating the root locus to the possible combinations of eigenvalues and invariant subspaces. Another possibility is the application of these two level periodic feedback controls to stable linear systems. After all, the electronic realization of this type of control is only slightly more complicated than constant output feedback gains or compensation networks.

ACKNOWLEDGEMENT

The authors thank John Montague for performing the analog computer simulation of the first example.

REFERENCES

[1] Brockett, R. W., and Lee, H. B. "Frequency-Domain Instability Criteria for Time-Varying and Non-Linear Systems," Proceedings of the IEEE, Vol. 55, No. 5, May 1967.

[2] Brockett, R. W., Finite Dimensional Linear Systems, John Wiley and Sons, Inc., New York, N. Y., 1970.

[3] Welch, R. V. "On the Design of Stable Linear Systems Using Two-Level Periodic Output Feedback Gains" M. S. Thesis, University of Maryland, June 1972.

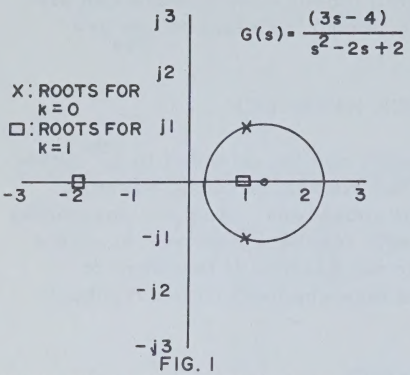


FIG. 1
ROOT LOCUS FOR SYSTEM (2.1) AS k VARIES FROM $-\infty$ TO $+\infty$.

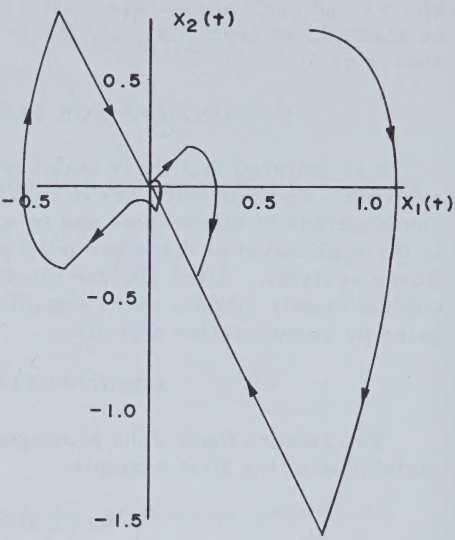


FIG. 2
PHASE PLANE TRAJECTORY FOR EXAMPLE 1.

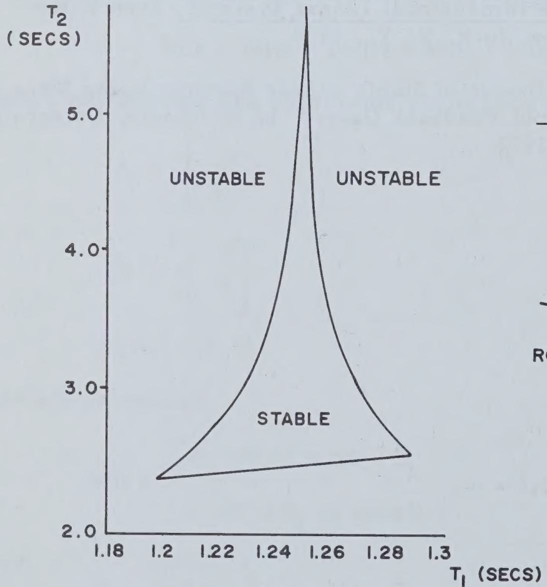


FIG. 3
STABILITY REGION IN T_1, T_2 COORDINATES FOR EX. 1.

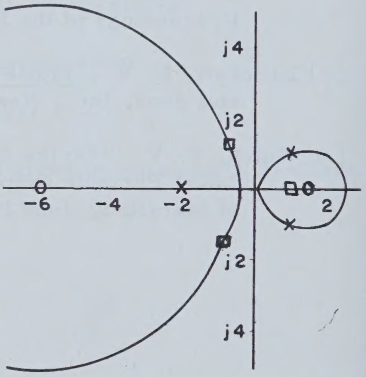


FIG. 4
ROOT LOCUS FOR THIRD ORDER SY.

X: ROOTS FOR k=0
□: ROOTS FOR k=1
O: ZEROES OF G(s)

$$G(s) = \frac{.75s^2-3.25s-7}{s^3-2s+4}$$

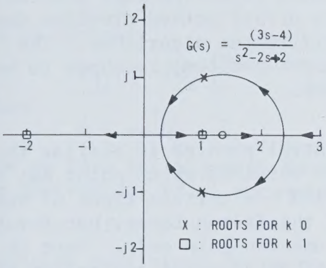


FIGURE 1. ROOT LOCUS FOR EXAMPLE 1.

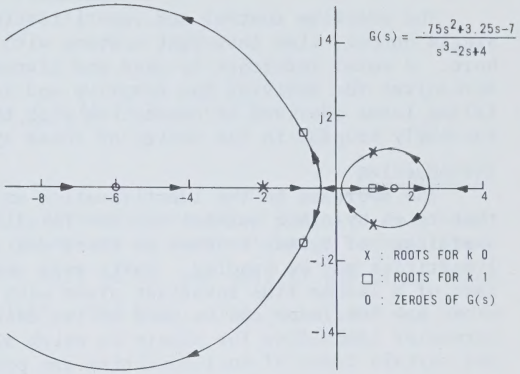


FIGURE 4. ROOT LOCUS FOR EXAMPLE 2
ARROWS IN DIRECTION OF INCREASING k.

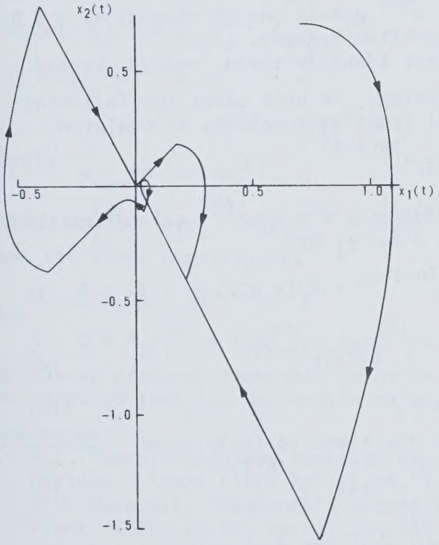


FIGURE 2. TYPICAL PHASE PLANE TRAJECTORY
FOR EXAMPLE I.

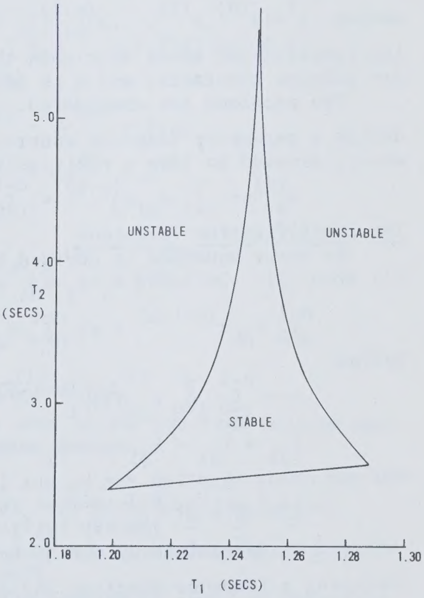


FIGURE 3. A STABILITY REGION IN T_1, T_2
COORDINATES FOR EXAMPLE I.

ADAPTIVE CONTROL AND IDENTIFICATION VIA LIAPUNOV'S DIRECT METHOD

RICHARD V. MONOPOLI
University of Massachusetts, Amherst, Massachusetts

ABSTRACT

The adaptive control and identification problems for single input, single output, time invariant systems with unknown parameters are considered here. A model reference is used and Liapunov's direct method provides the mechanisms for deriving the adaptive and identification algorithms. The Kalman Lemma (derived in connection with the Lure' problem) is shown to be extremely helpful in the design of these systems.

Introduction

The approach to the identification and control problem is similar to that taken by other authors but new results are obtained which point out the usefulness of Kalman's lemma in these designs and how certain types of nonlinearities may be handled. Parks made use of the Kalman Lemma, but for the case of a linear time invariant plant with an unknown gain only. Here it is shown how the lemma can be used in the design of an adaptive controller or a parameter identifier for plants in which all of the parameters are unknown and certain types of nonlinearities are present.

Statement of the Problem

A dynamic system is described by the differential equation

$$\sum_{i=0}^{n} a_i x^{(n-i)} = \sum_{j=0}^{n-1} \sum_{i=0}^{m} b_{ji} f_i(\underline{x},u,t)^{(n-1-j)} \qquad (1)$$

where $a_0 = 1$, superscript is i^{th} derivative w.r.t. time, $x^{(0)}$ is plant output, $\underline{x}^T = (x^{(0)}, x^{(1)}, .., x^{(n-1)})$, $f_i(\underline{x},u,t)$ are nonlinear and/or time varying functions of known form such that $x^{(0)}$ exists and is unique, a_i and b_i are unknown constants, and m is any positive integer.

Two problems are considered. First identify the a_i and b_i, second design a parameter adaptive control system. In both cases the following model, assumed to have a positive real transfer function, is employed.

$$x_d^{(n)} + \sum_{i=1}^{n} a_{di} x_d^{(n-i)} = \sum_{i=0}^{n-1} b_{di} w^{(n-i-1)} \qquad (2)$$

The Identification Technique

An error equation is derived by defining $e = x_d - x^{(0)}$ and subtracting (1) from (2). Defining $\Delta_j = a_j - a_{dj}$ and z and y_i by:

$$\sum_{h=0}^{n-1} b_{dh} z^{(n-1-h)} = x; \quad \sum_{h=0}^{n-1} b_{dh} y_i^{(n-1-h)} = f_i(\underline{x},u,t) \qquad (3)$$

Define:

$$w = \sum_{j=0}^{n-1} \sum_{i=0}^{m} k_{ji}(t) y_i^{(n-1-j)} - \sum_{i=1}^{n} k_i(t) z^{(n-i)} \qquad (4)$$

$$\delta_{ji} = k_{ji} - b_{ji}\ ; \qquad \delta_i = \Delta_i - k_i \qquad (5)$$

Now the error equation may be put into the state variable form

$$\dot{\underline{e}} = A\underline{e} + \underline{d}\Big(\sum_{j=0}^{n-1} \sum_{i=0}^{m} \delta_{ji} y_i^{(n-1-j)} + \sum_{i=1}^{n} \delta_i z^{(n-i)} \Big) \qquad (6)$$

where $\underline{d}$ is determined by the a_i and b_{di}.

Defining a Liapunov function and calculating its derivative gives:

$$V = \underline{e}^T P \underline{e} + \sum_{j=0}^{n-1} \sum_{i=0}^{m} \delta_{ji}^2 + \sum_{i=1}^{n} \delta_i^2 \qquad (7)$$

$$\dot{V} = -\underline{e}^T Q \underline{e} + \sum_{j=0}^{n-1} \sum_{i=0}^{m} (2e_1 y_i^{(n-1-j)} + \dot{k}_{ji}) \delta_{ji} + \sum_{i=1}^{n} (2e_1 z^{(n-1-i)} + \dot{k}_i) \delta_i \qquad (8)$$

where $e_1 = e$ the output error, $-Q = A^T P + PA$, and Q is chosen via Kalman's Lemma such that

$$\underline{e}^T P \underline{d} = e_1$$

If we define:

$$\dot{k}_{ji} = -2e_1 y_i^{(n-1-j)} \quad ; \quad \dot{k}_i = -2e_1 z^{(n-1-i)} \qquad (9)$$

Then:

$$\dot{V} = -\underline{e}^T Q \underline{e} \qquad (10)$$

Now we have the Liapunov function's derivative negative definite in $\underline{e}$ and no derivatives of e_1 are needed. Equations (9) insure that $\underline{e} \rightarrow \underline{0}$ as time goes to infinity but the δ_{ji} and δ_i will go to zero only if the plant input contains sufficient frequency content. See Lion, AIAA, Oct. 1967 for a discussion of convergence.

Parameter Adaptive Control

The parameter adaptive control technique presented here is very similar to the identification technique above. The plant is assumed to have all its poles and zeros in the open left half plane. The input to the model is some reference input, r(t), and the control input, u(t), is defined in terms of the adaptive parameters. The development is sketched out below for a second order example.

Plant: $x^{(2)} + a_1 x^{(1)} + a_2 x^{(0)} = b_1 u^{(1)} + b_2 u^{(0)} + b_3 f(x) \quad b_1 \neq 0$

Model: $x_d^{(2)} + a_{1d} x_d^{(1)} + a_{2d} x^{(0)} = b_{2d} r(t)$

Define: $y_1^{(1)} + c y_1^{(0)} = u^{(0)} \quad ; \quad y_2^{(1)} + c y_2^{(0)} = f(x)$

$z_1^{(1)} + c z_1^{(0)} = r(t) \quad ; \quad z_2^{(1)} + c z_2^{(0)} = x^{(0)}$

Then: $u = y_1^{(1)} + c y_1^{(0)}$

Let: $y_1^{(1)} = k_1 y_1^{(0)} + k_2 y_2^{(0)} + k_3 z_1^{(0)} + k_4 z_2^{(1)} + k_5 z_2^{(0)}$

Then the error equation is:

$$\dot{\underline{e}} = A\underline{e} + \underline{d}\zeta$$

With

$$\zeta = \delta_1 y_1^{(0)} + \delta_2 y_2^{(0)} + \delta_3 z_1^{(0)} + \delta_4 z_2^{(1)} + \delta_5 z_2^{(0)}$$

The theory proceeds from this point as it does in the previous section and the adaptive laws are derived in an analogous manner.

References:

1. P.C. Parks, "Liapunov Redesign of Model Reference Adaptive Control Systems," Trans. IEEE AC-11, No. 3, 362-367 (1966).
2. R.V. Monopoli, "Liapunov's Method for Adaptive Control System Design," Trans. IEEE, AC-12, No. 3, 334-335 (1967).
3. R.E. Kalman, "Liapunov functions for the problem of Lure' in Automatic Control," Proc. of National Academic Science, vol. 49, 201-205 (1963).
4. P.M. Lion, "Rapid Identification of Linear and Nonlinear Systems," JACC Proceedings, Seattle, Washington, pp. 605-615 (1966).

Acknowledgement

This research was supported in part by the National Aeronautics and Space Administration under grant NGL-22-010-018.

DUAL VARIATIONAL TECHNIQUES AND THE EPSILON METHOD

W. L. CHAN M. D. LODAYA D. J. EWING, JR.

The University of Toledo
Toledo, Ohio 43606

ABSTRACT

Dual variational formulations of control problems lead to duality and measure of suboptimality. Balakrishnan's epsilon method is applied to investigate these aspects numerically in two specific optimal control problems; one described by a linear partial differential equation and the other by a nonlinear ordinary differential equation. The primal and the dual systems of both problems are optimized independently while the respective functionals furnish computable, a priori, measures of suboptimality at various stages of the calculations.

INTRODUCTION

Numerical applications of dual formulations of control problems have been started by Pearson [1] and Ringlee [2]. More recent investigations have been carried out by Ewing who employed the general formulation of complementary (dual) variational principles advanced by Noble [3] to obtain suboptimal measure for both lumped and distributed linear systems. Computational studies by Ewing include analog [4], digital [5] as well as hybrid [6]. In this work, the computational aspects of duality and suboptimality are examined by adopting an alternative approach to complementary variational techniques in control, found in Chan [7]. In this approach, a dual system is constructed once the primal system is given, and the two can be optimized separately when direct methods are applied. Furthermore, the primal functional always acts as an upper bound to the dual functional which, in turn, acts as a lower bound to the former. They are equal at the optimal solution. Hence, they provide a computable suboptimal measure a priori. These features are demonstrated numerically for two optimal control problems by employing Balakrishnan's epsilon method [8] which is frequently employed in the current literature [9,10]. The epsilon method, while avoiding the solutions of the usual two-point boundary-value problems, as required by the necessary conditions for optimality, enables the primal and dual systems to be optimized independently of each other, so that numerical difficulties arising in one may not occur in the other, as shown by the following solutions.

DUAL VARIATIONAL PRINCIPLE

Let H and H' be two real Hilbert spaces. $J(x)$, defined on H, is a real-valued, bounded, Frechet differentiable, convex functional and L is a closed dense linear operator such that $L:H \to H'$. Given the primal problem of finding $\overline{x}$ such that

$$J(\overline{x}) = \operatorname*{Min}_{x} J(x) \text{ , subject to } Lx = 0 \text{ ,} \tag{1}$$

following [7], the dual problem can be constructed as the determination of $\overline{y}$ such that

$$I(\overline{y}) = \operatorname*{Max}_{y} I(y) \tag{2}$$

subject to $\delta J(y)h = \langle h, L^*p\rangle$ for all $h \varepsilon H$, (3)

where $I(y) \equiv J(y) - \langle L^*p, y\rangle$. L^* is the adjoint of L, $p \varepsilon D(L^*)$ the domain of L^* and $\delta J(y)$ is the Frechet derivative of J at y. If $\overline{x}$ is the solution to (1) then $\overline{x} = y$ and, further,

$$J(x) \geq J(\overline{x}) = I(\overline{x}) \geq I(x) \quad \text{for all } x \varepsilon H.$$

Expressions (2), (3) and (4) constitute the dual (complementary) variational principle which gives a dual characterization of the original optimization problem: the primal problem defined by (1) and the dual defined by (2) and (3). The solution of the primal problem is also a solution to the dual problem and vice versa. Therefore, for an arbitrary $x \varepsilon H$, (4) provides a measure of suboptimality

$$J_{sub}(x) = J(x) - I(x)$$

which is equal to zero at $x = \overline{x}$. Note that this index is independent of any a priori information concerning the optimal solution.

THE EPSILON METHOD

One way of obtaining independent numerical solutions to the primal and dual optimization problems is to apply Balakrishnan's ε-method. The gist of this method is as follows. Consider systems governed by equations of the form

$$x(t) = A\,x(t) + B\,u(t)\ , \quad x(0) = 0\ , \tag{5}$$

where for each $t \varepsilon [0,T]$, the state $x(t)$ and the control $u(t)$ are elements of the Hilbert spaces H_1 and H_2 respectively. A is a linear operation mapping a domain $D(A)$, dense in H_1, into H_1, and B is a bounded linear operator mapping H_2 into H_1.

Given the system governed by (5), a set $U \varepsilon L^2(T,H_2)$ of admissible controls and a cost functional $J(u,x)$, it is required to find $\overline{u} \varepsilon U$, $\overline{x} \varepsilon D(L)$ such that

$$J(\overline{u},\overline{x}) = \inf_{u \varepsilon U\ ;\ Lx = Bu} J(u,x) \quad \text{where} \quad L \equiv \frac{\partial}{\partial t} - A\ . \tag{6}$$

The ε-method, being an extension of Courant's penalty function method, converts the optimization problem (6), with the dynamic constraint (5), to an unconstrained one in the following way.

Let a functional J_ε, the ε-functional, be defined by

$$J_\varepsilon(u,x) = J(u,x) + \frac{1}{\varepsilon}\,||Lx - Bu||^2$$

where $\varepsilon > 0$, and the norm is the $L^2(T;H_1)$ norm. The problem is that of finding $u_\varepsilon \varepsilon U$, $x_\varepsilon \varepsilon D(L)$ such that

$$J_\varepsilon(u_\varepsilon,x_\varepsilon) = \inf_{u \varepsilon U,\ x \varepsilon D} J_\varepsilon(u,x)\ . \tag{7}$$

Balakrishnan [8] and DeJulio [9] have proven that, as ε approaches zero, u_ε and x_ε approach $\overline{u}$ and $\overline{x}$ respectively, thus solving the original problem with dynamic constraint.

A Distributed System for Tubular Reactors [11]

Given the problem of minimizing $J_p(u)$, where

$$J_p(u) = \frac{1}{2}\int_0^1\int_0^1 [\mu^2 x^2(r,t) + u^2(r,t)]drdt \tag{8}$$

Subject to $\frac{\partial x}{\partial t} + \frac{\partial x}{\partial r} = u(r,t)$; $t \varepsilon [0,1]$, $r \varepsilon [0,1]$,

$x(r,0) = r$, $x(0,t) = 0$, $\mu = 5$,

then, according to the dual variational principle (for details see [5,7]), the dual problem is maximization of $J_d(x)$ where

$$J_d(x) = -\frac{1}{2}\int_0^1\int_0^1 (\mu^2 x^2 + p^2)drdt + \int_0^1 rp(r,0)dr \tag{9}$$

subject to $\frac{\partial p}{\partial t} + \frac{\partial p}{\partial r} = -\mu^2 x$

$p(1,0) = 0$; $p(r,1) = 0$.

Following (8), the ε-functionals of (8) and (9) are given by

$$J_{p\varepsilon} = \frac{1}{2}\int_0^1\int_0^1 [\mu^2 x^2(r,t) + u^2(r,t)]drdt + \frac{1}{2\varepsilon}\int_0^1\int_0^1 (\frac{\partial x}{\partial t} + \frac{\partial x}{\partial r} - u)^2 drdt \,, \tag{8ε}$$

and

$$J_{d\varepsilon} = -\frac{1}{2}\int_0^1\int_0^1 [\mu^2 x^2(r,t) + p^2(r,t)]drdt + \int_0^1 rp(r,0)dr + \frac{1}{2}\int_0^1\int_0^1 (\frac{\partial p}{\partial t} + \frac{\partial p}{\partial r} + \mu^2 x)^2 drdt \,. \tag{9ε}$$

A Nonlinear Lumped System [12]

The primal problem is the minimization of $I_p(u)$ where

$$I_p(u) = \frac{1}{2}\int_0^1 (x^2 + u^2)dt \tag{10}$$

subject to $\dot{x} = -x^2 + u$, $x(0) = x$.

The dual problem [7] is maximization of $I_d(x)$ where

$$I_d(x) = \int_0^1 (-\frac{x^2}{2} - \frac{p^2}{2} + 3x^2 p) + p(0)x_o \tag{11}$$

subject to $\dot{p} = 2xp + x$, $p(1) = 0$.

The ε-functionals, in this case, are

$$I_{p\varepsilon} = \frac{1}{2}\int_0^1 (x^2 + u^2)dt + \frac{1}{2\varepsilon}\int_0^1 (x + x^2 - u)^2 dt \qquad (10\varepsilon)$$

and

$$I_{d\varepsilon} = \frac{1}{2}\int_0^1 \left(-\frac{x^2}{2} - \frac{p^2}{2} + 3\,x^2\,p\right) + p(0)x_o$$
$$+ \frac{1}{2\varepsilon}\int_0^1 (\dot{p} - 2\,x\,p - x)^2 dt\ . \qquad (11\varepsilon)$$

These ε-functionals (9ε), (8ε), (10ε) and (11ε) are discretized by application of a central difference scheme and are optimizated by the following iterative procedures:

1. An initial trial function, meeting the required boundary conditions x, is selected.

2. u is computed by setting the partial derivative of J_ε w.r.t. u equal to zero.

3. The minimizing value of x corresponding to the u evaluated in 2 is calculated by means of a conjugate gradient technique.

4. 2 and 3 are repeated until convergence is achieved (or a pre-assigned tolerance of J_{sub} is reached).

These procedures have been implemented on an IBM 360/44 computer. Suboptimal measures and the bounding effects are illustrated through plots of the primal and dual functionals. In Fig. 1 $J_{p\varepsilon}$ and $J_{d\varepsilon}$ are plotted for increasing numbers of iterations and various values of ε. To demonstrate the effect of discretization only, exact solutions of (8), from [11], have been used. Fig. 2 shows that J_d, indeed, acts as a lower bound to J_p. This demonstrates the fact that J_{sub}, obtained from the dual variational principle, is independent of the way in which $\overline{x},\overline{u}$ are determined. Fig. 3 is the counterpart of Fig. 1 for $I_{p\varepsilon}$ and $I_{d\varepsilon}$. Profiles of the optimal states and controls for both problems are shown in Fig. 4,5 and 6,7. They are in agreement with either exact or approximate solutions obtained by other methods previously reported [12,13].

In problem 1, optimization of the primal system requires less computer time. Computation of the control law is less critically dependent on initial values of the trial function when the dual formulation is applied to the system of problem 2. This indicates, in addition to suboptimal measures, that duality may offer an attractive alternative in obtaining numerical solutions. This alternative is valuable when computational difficulties occur in one but not in the other.

CONCLUSIONS

Two computational aspects of dual variational formulations have been demonstrated through computation of optimal solutions for a distributed linear system and for a lumped nonlinear system. These are: (1) suboptimal measures can be evaluated at any stage of the calculation, and (2) the primal and dual problems can be optimized independently by applying the ε-method, which has been shown to be a valuable numerical technique in dynamic system optimization.

REFERENCES

1. J. D. Pearson, "A Successive Approximation Method," Fourth Joint Automatic Control Conference, Minneapolis, Minn., June 1963, p. 36-41.

2. A. V. Ringlee, "Bounds for Convex Variational Programming Problems Arising in Power System Scheduling and Control," IEEE Trans. Automat. Contr., Vol. AC-10, No. 1, pp. 28-35, 1965.

3. B. Noble, "Complementary Variational Principles for Boundary-Value Problems, I. Basic Principles," Report No. 473, Math. Research Centre, University of Wisconsin, 1964.

4. D. J. Ewing, Jr., "Reciprocal Variational Techniques and the Solution of Optimization Problems," SIMULATION, Vol. 11, pp. 253-257, 1968.

5. D. J. Ewing, Jr. and T. J. Higgins, "Duality: and the Suboptimal Control of Distributed-Parameter Systems," PROCEEDINGS, IFAC Symposium on the Control of Distributed Parameter Systems, Banff, Alberta, Canada, June 21-23, 1971, paper 3-5.

6. D. J. Ewing, Jr., "A Hybrid Computer Design Technique for Suboptimal Controllers," Summer Computer Simulation Conference, San Diego, California, June 14-16, 1972, pp. 374-381.

7. W. L. Chan, "Duality in Optimal Control," Ph.D. Dissertation (in preparation), The University of Toledo, Toledo, Ohio.

8. A. V. Balakrishnan, "On a New Computing Technique in Optimal Control," SIAM J. Contr., 6, 1968, pp. 149-173.

9. S. De Julio, "Numerical Solution of Dynamical Optimization Problems," SIAM J. Contr., 8, 1970, pp. 135-147.

10. J. M. Taylor and C. T. Constantinides, "Optimization: Application of The Epsilon Method," IEEE Trans. Automat. Contr., Vol. AC-17, No. 1, pp. 128-131, 1972.

11. L. B. Koppel and Y. P. Shih, "Optimal Control of a Class of Distributed-Parameter Systems with Distributed Controls," Ind. Eng. Chem. Fundamentals, Vol. 7, No. 3, 1968, pp. 414-422.

12. A. P. Sage, Optimum System Control, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1968.

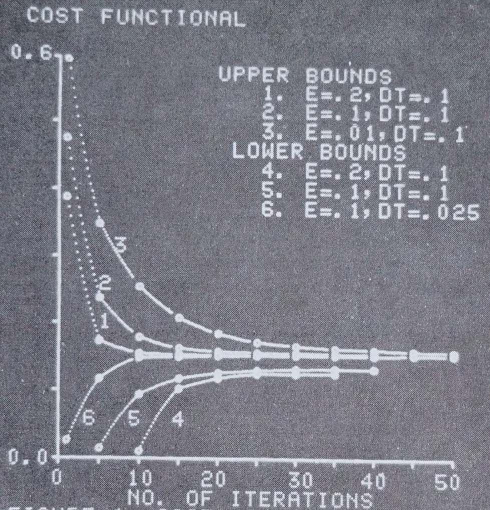


FIGURE 1: COST VS. ITERATION NO., DISTRIBUTED SYSTEM.

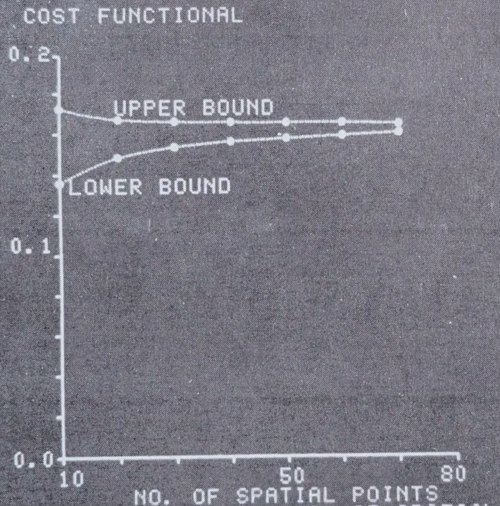


FIGURE 2: COST VS. NO. OF SPATIAL POINTS IN APPROXIMATION.

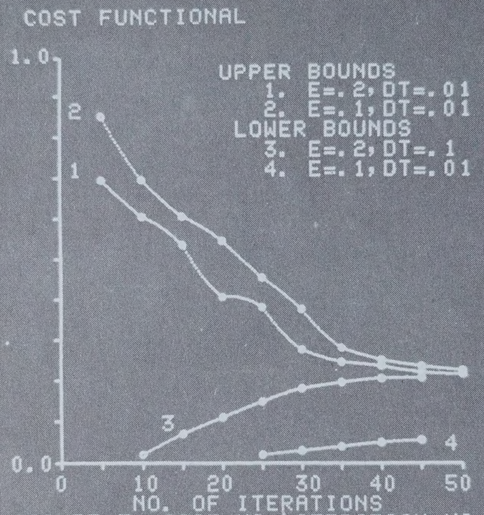


FIGURE 3: COST VS. ITERATION NO., NONLINEAR SYSTEM.

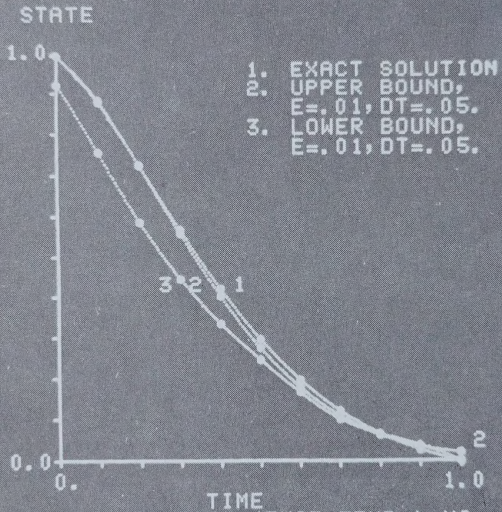


FIGURE 4: STATE (EXIT TEMP.) VS. TIME, DISTRIBUTED SYSTEM.

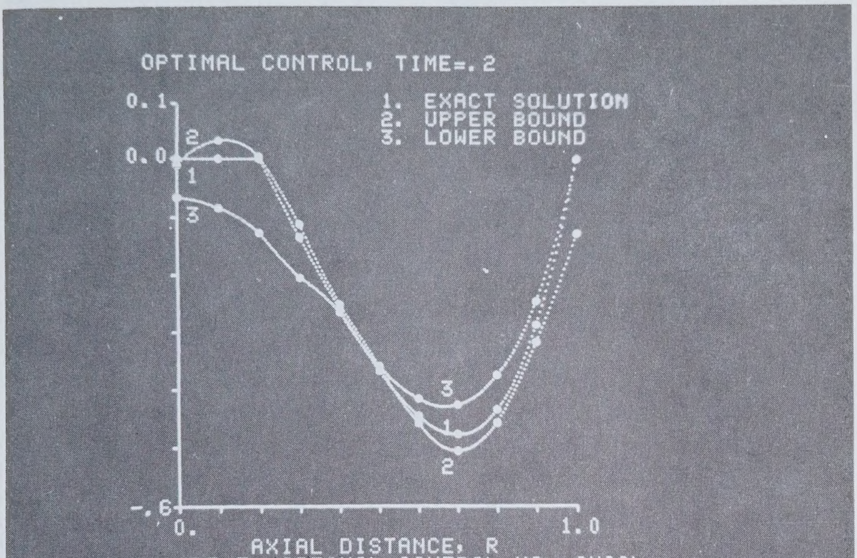


FIGURE 5: OPTIMAL CONTROL VS. AXIAL DIST, DISTRIBUTED SYSTEM.

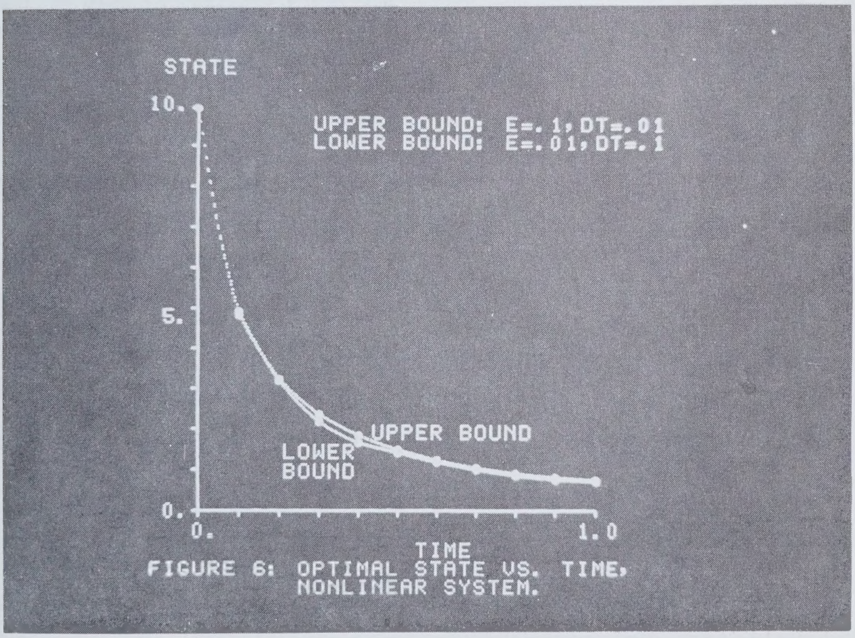


FIGURE 6: OPTIMAL STATE VS. TIME, NONLINEAR SYSTEM.

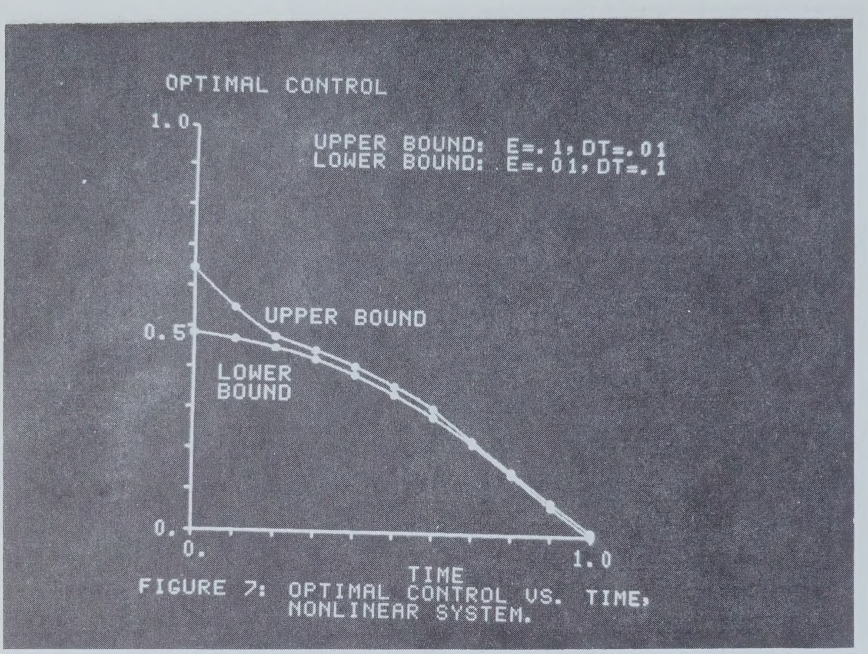


FIGURE 7: OPTIMAL CONTROL VS. TIME, NONLINEAR SYSTEM.

AIRCRAFT PARAMETER IDENTIFICATION: A VARIATIONAL PROCEDURE*

R.E. BACH, JR.
NORTHEASTERN U.
BOSTON, MA

SUSAN E. POST
U.S. ARMY AMRDL
MOFFETT FIELD, CA

LT. J.E. TUCKER
NAVAL POSTGRADUATE S.
MONTEREY, CA

The determination of aircraft stability and control derivatives from flight-test data continues to be an important application of parameter-estimation theory. The identification problem, of course is much the same in all fields, although few physical models with typical process and measurement disturbances are as well understood as the aerodynamic model. An estimation procedure, applied to aircraft parameter identification, has been described recently by Mehra, et al [1]. The method includes identification of process and measurement-noise statistics, Kalman filter gains and system model parameters, and employs a maximum-likelihood criterion. In this paper we present a smoothing algorithm that appears somewhat simpler. [2].

Consider a continuous system model

$$\dot{x} = f(x,u,p) + w, \quad x(t_o) = x_o; \quad z = h(x,u) + v, \tag{1}$$

where x is the (n x 1) state vector, u is the (m x 1) input vector, p is a constant (k x 1) parameter vector, w is an (n x 1) vector that includes process noise and modeling errors, z is the (r x 1) output vector, and v is an (r x 1) vector of measurement noise. We assume that w and v are stationary, uncorrelated white-noise processes, each having zero mean with covariance

$$E[w(t)w(\tau)] = Q\ \delta(t-\tau); \quad E[v(t)v(\tau) = R\ \delta(t-\tau). \tag{2}$$

We wish to estimate x,p,w,Q,R, given measurements of u, z on the interval (t_o,t_f) such that the criterion

$$J = \int_{t_o}^{t_f} (w^T Q^{-1} w + v^T R^{-1} v)\ dt/2 + (\ln |R| + \ln |Q|)/2 \tag{3}$$

is minimized. If we adjoin the differential equation (1) to the criterion (3) with Lagrange multiplier λ in the usual way, we obtain for variations in x,p,w,Q,R, the variation

$$\begin{aligned}\delta\bar{J} &= (\partial\bar{J}/\partial Q)\delta Q + (\partial\bar{J}/\partial R)\delta R + \lambda^T(t_o)\delta x_o - \lambda^T(t_f)\delta x_f \\ &+ \int_{t_o}^{t_f} [(\partial H/\partial x + \dot{\lambda}^T)\delta x + (\partial H/\partial p)\delta p + (\partial H/\partial w)\partial w]dt,\end{aligned} \tag{4}$$

where the Hamiltonian H is defined to be

$$H = (w^T Q^{-1} w + v^T R^{-1} v)/2 + \lambda^T(f + w). \tag{5}$$

Now, if in (4) we set

$$\begin{aligned}&\partial\bar{J}/\partial Q = 0\ ;\ \partial\bar{J}/\partial R = 0\ ;\ \partial H/\partial w = 0; \\ &\dot{\lambda}^T = -\partial H/\partial x; \ \lambda^T(t_f) = 0,\end{aligned} \tag{6}$$

---

*Research performed during ASEE-NASA Summer Faculty Institute 1972.

then the variation in the criterion reduces to

$$\delta\bar{J} = \lambda^T(t_o)\ \delta x_o + \int_{t_o}^{t_f} (\partial H/\partial p)\ dt\ \delta p. \tag{7}$$

Hence, the 1 x (n + k) gradient vector is given by

$$(\partial\bar{J}/\partial x_o,\ \partial\bar{J}/\partial p) = [\lambda^T(t_o),\ \int_{t_o}^{t_f} (\partial H/\partial p)dt]. \tag{8}$$

It can be shown that the first three relations of (6) lead to

$$R = \int_{t_o}^{t_f} vv^T dt \quad ; \quad Q = \int_{t_o}^{t_f} ww^T dt \quad ; w = -Q\lambda. \tag{9}$$

Note that if (9) is used in (3), the criterion becomes

$$J = (r + n)/2 + (\ln |Q| + \ln |R|)/2, \tag{10}$$

so that a convergent iterative procedure minimizes

$$J_a = (\ln|Q||R|)/2. \tag{11}$$

We can now formulate a strategy for estimation of the states, parameters, and noise statistics:

(1) Integrate the state equation (1) forward (use $w = 0$ and guess of x_o, p for the first iteration). Compute R from (9).

(2) Integrate the costate equation (6) backward, using v obtained during the forward integration. Evaluate the gradient from (8) and Q from (9).

(3) Take a step δx_o, δp "downhill". Each iteration after the first will be made with $\overset{o}{w}$ from (9), using the λ calculated during the backward integration of the previous iteration. Continue until a minimum of J_a is reached.

A parameter-identification algorithm, based on a variational optimization procedure, has been described. No mention of important matters as, identifiability, correlation of measurement and process noise, error-free channels, or convergence has been made in this paper. At this writing, preliminary experiments with aircraft flight data are encouraging, but no conclusive results are yet available.

[1] Mehra, R.K.; Stepner, D.E.; Tyler, J.S., "A Generalized Method for the Identification of Aircraft Stability and Control Derivatives from Flight-Test Data", JACC, August 1972.

[2] Bryson, A.E., Jr.; Ho, Y.C., Applied Optimal Control, Blaisdell, 1969, Ch. 13.

ON-LINE SYSTEM IDENTIFICATION

NARESH K. SINHA
Department of Electrical Engineering and
Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada

Abstract The problem of identification of the parameters of a continuous system from samples of observed input and output data is considered. From the samples a discrete-time model is first derived using a recursive algorithm so that computations for updating the model are completed during the sampling interval. From this model, a continuous system model is then obtained. Applications of the method to the identification of the parameters of reactors is described.

INTRODUCTION

Adaptive control of a physical plant subject to large variations in parameters would be considerably simplified if there were available a method for rapid identification of the mathematical model of the plant. A number of methods of identification have been proposed in the literature(1-5) but they usually require a considerable amount of computation, in addition to the fact that one has to wait for the accumulation of sufficient data to make the computations possible. Moreover, a scheme for identification is of little practical value if it takes a long time to determine the model parameters, since by then these might already have changed. There is thus a need for a method of online identification which allows one to start with an estimate of the parameters of the model even after the first set of data has been obtained, and then to update this estimate using a recursive technique as more data arrive. Furthermore, a realistic method must take into account the fact that all measurements are contaminated with noise.

As a digital computer is generally employed for online identification, one must of necessity use samples of input and output data. The problem can, therefore, be conveniently divided into two parts. The first part consists of determining a discrete-time model for the system from the sampled data. The online methods for doing this can be put into four main classes, (i) those which are based on sequential regression methods(6,7), (ii) those which are based on correlation function techniques(8), (iii) those which are based on stochastic approximation(9,10), and (iv) those which are based on sequential learning(11). The second part of the problem is the estimation of the parameters of the transfer function of the continuous system from the discrete model(12,13).

The object of this paper is to discuss the application of some of these techniques to the online identification of the parameters of a physical system. First a brief survey of the methods for online identification will be presented, with a discussion of their relative advantages. The results of the application of one of these methods to the identification of the parameters of a nuclear reactor from noisy input-output data will then be presented.

ON-LINE IDENTIFICATION OF A DISCRETE-TIME SYSTEM

In this section, various methods which have been proposed for online identification of discrete-time systems will be considered. For the sake of simplicity, only single-input single-output systems will be discussed, although extension to the multivariable case for each method is straightforward.

Consider a discrete-time linear system described by the difference equation

$$a_o r_i + a_1 r_{i-1} + \ldots + a_m r_{i-m} = c_i + b_1 c_{i-1} + \ldots + b_n c_{i-n} \tag{1}$$

where $r_i = r(iT)$ is the input sequence, $c_i = c(iT)$ is the output sequence, T is the sampling interval, and n is greater than m.

The determination of the parameters $a_o, a_1, \ldots, a_m; b_1, b_2, \ldots, b_n;$ is straightforward in the absence of noise, as these can be determined by solving $p \triangleq m+n+1$ linear simultaneous equations in terms of the input-output data. However, due to noise, instead of r_i, and c_i, measurements are available of u_i and y_i, defined as

$$u_i = r_i + w_i \tag{2}$$

$$y_i = c_i + v_i \tag{3}$$

where w_i and v_i are uncorrelated and zero-mean noise sequences. Define the parameter vector

$$\underline{\phi} \triangleq \begin{bmatrix} a_o & a_1 & a_2 & \ldots & a_m & b_1 & b_2 & \ldots & b_n \end{bmatrix}^T \tag{4}$$

where the superscript T represents the transpose. Then a noisy estimate $\underline{\phi}_i$ of $\underline{\phi}$ is obtained by solving the matrix equation

$$A_i \underline{\phi}_i = \underline{z}_i \tag{5}$$

$$\text{where } A_i \triangleq \begin{bmatrix} u_i & u_{i-1} \cdots\cdots u_{i-m} & -y_{i-1} & -y_{i-2} \cdots -y_{i-n} \\ u_{i+1} & u_i \cdots\cdots u_{i-m+1} & -y_i & -y_{i-1} \quad -y_{i-n+1} \\ \vdots & \vdots \qquad \vdots & \vdots & \vdots \qquad \vdots \\ u_{i+p-1} & u_{i+p-2} \quad u_{i-m+p-1} & -y_{i+p-2} & -y_{i+p-3} \quad -y_{i-n+p-1} \end{bmatrix} \tag{6}$$

and $$z_i \triangleq \left[y_i \; y_{i+1} \; \cdots \; y_{i+p-1} \right]^T \tag{7}$$

Assuming that A_i is nonsingular, it can be inverted to give

$$\underline{\phi}_i = A_i^{-1} \; \underline{z}_i \tag{8}$$

As this provides a noisy estimate of $\underline{\phi}$, one would need more input-output data to get a better estimate. Moreover, as matrix inversion is generally time-consuming, this approach is not suitable for "on-line" identification. With more input-output data, one may add further rows to the matrix A_i, and the vector $\underline{z}_i$ to obtain a least-squares estimate of the parameter vector given by

$$\hat{\underline{\phi}}_k = A_k^+ \; \underline{z}_k \tag{9}$$

where A_k^+ is the pseudoinverse of A_k, defined as

$$A_k^+ = A_k^T \left[A_k \, A_k^T \right]^{-1} \quad \text{for } k \leq p \tag{10}$$

and $$A_k^+ = \left[A_k^T \, A_k \right]^{-1} A_k^T \quad \text{for } k > p \tag{11}$$

The pseudoinverses defined by eqns. (10) and (11), provide a minimum-norm and a least-square solution for the parameter vector, respectively. They exist and are unique if the matrix A_k possesses full rank, defined as equal to the number of rows or columns, whichever is smaller. This condition is assured if the input sequence r_i satisfies any of the following set of three conditions:

(i) r_i is a random sequence

(ii) $r_i = 0$ for $i < n$
$ = 1$ for $i > n$

(iii) r_i is a sequence composed of n discrete Fourier components and all natural modes are present in the output sequence.

At least one of these conditions is normally satisfied, but equation (11) is still unsuited for on-line identification due to the matrix inversion indicated. This difficulty is overcome by using a recursive algorithm for the determination of the pseudoinverse.

Let

$$A_{k+1} = \begin{bmatrix} A_k \\ \\ \underline{a}^T_{k+1} \end{bmatrix} \tag{12}$$

where $$\underline{a}_{k+1} = \left[u_{i+k} \quad u_{i+k-1} \cdots u_{i-m+k} \quad -y_{i+k-1} \quad -y_{i+k-2} \quad -y_{i-n+k} \right]^T \tag{13}$$

and let

$$\underline{z}_{k+1} = \begin{bmatrix} \underline{z}_k \\ y_{i+k} \end{bmatrix} \tag{14}$$

Then the following recursive algorithms may be derived$^{(7)}$ for online estimation of the parameter vector.

For $k \leq p$ (minimum norm solutions)

$$\hat{\underline{\phi}}_{k+1} = \hat{\underline{\phi}}_k + \frac{Q_k \underline{a}_{k+1}}{\underline{a}^T_{k+1} Q_k \underline{a}_{k+1}} \left(y_{i+k} - \underline{a}^T_{k+1} \hat{\underline{\phi}}_k\right) \tag{15}$$

$$\text{where } Q_{k+1} = Q_k - \frac{\left[Q_k \underline{a}_{k+1}\right] \left[Q_k \underline{a}_{k+1}\right]^T}{\underline{a}^T_{k+1} Q_k \underline{a}_{k+1}} \tag{16}$$

$$\text{and } P_{k+1} = P_k - \frac{\left[P_k \underline{a}_{k+1}\right] \left[Q_k \underline{a}_{k+1}\right]^T + \left[Q_k \underline{a}_{k+1}\right] \left[P_k \underline{a}_{k+1}\right]^T}{\underline{a}^T_{k+1} Q_k \underline{a}_{k+1}}$$

$$+ \frac{\left[Q_k \underline{a}_{k+1}\right] \left[Q_k \underline{a}_{k+1}\right]^T \left(1 + \underline{a}^T_{k+1} P_k \underline{a}_{k+1}\right)}{\left(\underline{a}^T_{k+1} Q_k \underline{a}_{k+1}\right)^2} \tag{17}$$

with the initial conditions

$$Q_o = I, \; P_o = 0 \text{ and } \hat{\underline{\phi}}_o = 0 \tag{18}$$

For $k > p$ (least-squares estimation)

$$\hat{\underline{\phi}}_{k+1} = \hat{\underline{\phi}}_k + \frac{P_k \underline{a}_{k+1} \left(y_{i+k} - \underline{a}^T_{k+1} \hat{\underline{\phi}}_k\right)}{1 + \underline{a}^T_{k+1} P_k \underline{a}_{k+1}} \tag{19}$$

$$\text{where } P_{k+1} = P_k - \frac{P_k \underline{a}_{k+1} \left[P_k \underline{a}_{k+1}\right]^T}{1 + \underline{a}^T_{k+1} P_k \underline{a}_{k+1}} \tag{20}$$

The matrices P_k and Q_k are shown to be

$$P_k = A_k^+ \cdot (A_k^+)^T \tag{21}$$

$$\text{and } Q_k = I - A_k^+ A_k \tag{22}$$

Thus, they are both symmetric. This fact, and the similarity of the two sets of equations further reduces the number of operations required in the formulation of the estimate $\hat{\underline{\phi}}$. The least squares estimate using equations (19) and (20) requires a total of

$$N = 3p^2 + 4p \tag{23}$$

multiplications. Moreover, the number of input-output data points required in storage at any stage k has been reduced from 2k + p-1 to p+1.

Applications of this algorithm to on-line identification have been presented in an earlier paper(7). The results are quite satisfactory for low noise, but in situations where the noise-to-signal level exceeded 10%, some difficulty was experienced. This is primarily due to the bias produced by the noise, and the results were found to be rather unsatisfactory when the noise level was more than 30%. Hence, some methods for making the estimates unbiased will now be considered.

One of these methods is based on correlation techniques. Mehra(8) has shown that estimates obtained using this method are asymptotically normal, unbiased and consistent. For instance, if for the system defined by eqn.(1) the input sequence, r_i, is a white noise sequence, an unbiased estimate of the parameters b_i is obtained from the following matrix equation

$$-\begin{bmatrix} \psi_1 & \psi_2 \cdots & \psi_n \\ \psi_2 & \psi_3 & \psi_{n+1} \\ \vdots & \vdots & \\ \psi_n & \psi_{n+1} & \psi_{2n-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} \psi_{n+1} \\ \psi_{n+2} \\ \vdots \\ \psi_{2n} \end{bmatrix} \tag{24}$$

$$\text{where } \psi_k \triangleq E\left[y_i \; y_{i-k}\right] \tag{25}$$

It can be shown that the matrix

$$\begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_n \\ \psi_2 & \psi_3 & & \psi_{n+1} \\ \vdots & \vdots & & \vdots \\ \psi_n & \psi_{n+1} & & \psi_{2n-1} \end{bmatrix}$$

known as the Hankel matrix for the system, is always nonsingular if the following condition is satisfied for all $j > 1$,

$$\psi_{n+j} = -\sum_{i=1} \psi_i \; b_{i+j-1} \tag{26}$$

In practice, the correlation terms must be obtained using the following relationship

$$\hat{\psi}_k = \frac{1}{N} \sum_{i=1}^{N} y_i \; y_{i-k} \tag{27}$$

where N is a large number. Hence, the method may involve considerable time for implementation. Some of this difficulty can be removed by using the recursive algorithm

$$\hat{\psi}_k^{N+1} = \hat{\psi}_k^N + \frac{1}{N+1} (y_{N+1}\, y_{N+1-k} - \hat{\psi}_k^N) \tag{28}$$

where $\hat{\psi}_k^{N+1}$ denotes the estimate of ψ_k based on (N+1) sample points.

The method discussed above can be extended to the general case of any input sequence, r_i, to yield an unbiased estimate of the parameter vector ϕ from estimates of the auto- and cross-correlation terms defined by $\sum y_i\, y_{i-k}$ and $\sum u_i\, y_{i-k}$ using recursive algorithms similar to eqn.(28). The inversion of the matrix can also be avoided by using an algorithm of the type of eqn. (15).

Other approaches to the problem of online identification are based on (a) stochastic approximation$^{(10)}$ and sequential learning$^{(11)}$. Algorithms of the former type may be written as

$$\hat{\underline{\phi}}_{k+1} = \hat{\underline{\phi}}_k - \rho_n\, \underline{\psi}_n \tag{29}$$

where ψ_n is a function that can be readily evaluated from the measurements of y_i and whose expected value is zero for accurate $\underline{\phi}$, and ρ_n is sequence of scalar correction coefficients. For convergence, the sequence ρ_n must satisfy certain conditions.

Sequential learning differs from the stochastic approximation approach in the derivation of its convergence features and in its convenient applicability to processes having slowly-time-varying parameters. The main advantage of sequential learning lies in the simple formulation of the algorithm, which is usually much faster than the correlation function method, and the well-known algorithms based on stochastic approximation. The implementation is still fairly involved.

DETERMINATION OF THE CONTINUOUS SYSTEM TRANSFER FUNCTION

If only the samples of the input-output data are known, the derivation of a unique transfer function from the discrete-time model is not possible. Smith$^{(12)}$ has proposed a method which can be used for determining the transfer function if the input function is known between the sampling instants as well. This, evidently, is not suitable for on-line identification, where the variation of the input at other than the sampling instants cannot generally be predicted, and can be measured only with considerable increase in hardware. Even then, the use of a digital computer makes this unrealistic. In a recent paper$^{(13)}$ it has been shown that if the sampling period T is selected such that $|p_k T| < 0.5$, where $|p_k|$ is the system pole farthest from the origin in the s-plane, the bilinear z transformation, described below, provides a very good approximation, besides being suitable from the computational point of view:

$$z^{-1} = \frac{2-sT}{2+sT} \tag{30}$$

$$s = \frac{2}{T}\,\frac{(1-z^{-1})}{(1+z^{-1})} \tag{31}$$

APPLICATION TO THE IDENTIFICATION OF THE PARAMETERS OF A NUCLEAR REACTOR

On-line identification of the parameters of a realistic model of a nuclear reactor will now be considered. The simplified block diagram representation of a typical nuclear reactor and essential parts of the power level controller are shown in Fig. 1. The operation of this system has been described in considerable detail in an earlier paper(14), and will not be discussed here. Although the system can be accurately represented by a set of ten first-order nonlinear differential equations, an incremental linear model is often adequate. As shown in a more recent paper(15), a second-order linear model can often be used effectively. To keep the identification problem fairly simple, such a model was assumed, and its parameters were determined using input-output data. The model has the following form

$$\frac{x(s)}{u(s)} = \frac{Ka_o}{s^2+a_1s+a_o} \tag{32}$$

Of the three parameters, K a_o and a_1, the forward gain K is adjustable and known, so that the identification problem consists of the online determination of a_o and a_1 from the samples of the input-output data. As the values of these parameters vary considerably with changes in the initial power-level, the temperature coefficient and the concentrations of the various delayed neutron groups, in order to be useful, the parameter identifications must be completed within a reasonably short time. These values are then used to determine the parameters of the feedback controller for a near-optimal control of the system. It will be seen, therefore, that a rapid identification of a_o and a_1 was essential for suitable control. This ruled out the application of the slower methods for on-line identification.

First, the regression method based on the recursive pseudoinverse algorithm was tried on a digital computer simulation of the reactor. The identification proceeded in two steps, as outlined earlier; i.e., first the parameters of the discrete model were obtained using this algorithm, and then the bilinear - z transform was used to determine the transfer function shown in eqn.(32). The sampling interval was taken as 0.1 second, and it was found that for the case of low noise (less than 10%), the values of the parameters converged very close to the actual values within one second. The computation and updating could be performed within the sampling interval. The variation of the estimates of a_1 and a_o with n are shown in Fig. 2. When the noise level was increased to about 30%, difficulties were experienced. In this case, it was found that the bias could be greatly reduced by using the method proposed by Sinha and Lawton(9) used in conjunction with an improved stochastic approximation algorithm(16). The time required for identification, however, was about 3.2 seconds, which is roughly three times that required for the previous method. With the noise level increased to more than 75%, this method did not converge in 10 seconds. In such a case, correlation methods may prove more useful, and a number of variations are being investigated.

CONCLUSION

A realistic approach to the online identification of systems has been presented with application to the determination of the parameters of a simplified model of a nuclear reactor control system. The main emphasis has been on the rapid determination of the model parameters for application in adaptive control. The investigations show that the recursive algorithm based on the pseudoinverse can be used very effectively for cases where the noise level was low. Increasing the noise level caused difficulties, which could be partly overcome by using an improved stochastic approximation algorithm together with a method which eliminated the need for repeated matirx inversion. For noise-to-signal ratios approaching one, however, correlation methods may be more appropriate though time-consuming.

REFERENCES

(1) Lee, R.C.K.: "Optimal Estimation, Identification and Control", M.I.T. Press, Cambridge, Massachussettes, 1964.

(2) Liff, A.I.: "System identification in the presence of noise by digital techniques", IEEE International Convention Record, 1966, vol. 14 (part 6), pp. 152-166.

(3) Eykhoff, P., Van der Grinten, P.M.E.M., Kwakernaak, H., and Veltman, B.P.T.: "Systems Modelling and Identification", Survey Paper, Proc. of

3rd IFAC Congress, London, 1966.

(4) Eykhoff,P.: "Process Parameter and State Estimation", Survey Paper 2, Proc. of IFAC Symposium on Identification, Prague, 1967.

(5) Balakrishnan, A.V. and Peterka,V.: "Identification in automatic control systems", Automatica, 1969, vol. 5, pp. 817-829.

(6) Hastings, James R. and Sage, M.W.: "Recursive generalized-least-square procedure for online identification of process parameters", Proceedings IEE, Vol. 116, December 1969, pp. 2057-2062.

(7) Sinha, N.K. and Pille,W.: "Online parameter estimation using the matrix pseudoinverse", Proc. IEE, vol. 118, Aug. 1971, pp. 1041-1046.

(8) Mehra, R.K.:"On-line identification of Linear Dynamic Systems with Applications to Kalman Filtering", IEEE Trans. on Automatic Control, vol. ac-16, Feb. 1971, pp. 12-21.

(9) Sinha,N.K. and Lawton, J.S.: "Unbiased estimates for on-line system identification", Proc. IEEE, vol. 59, November 1971, pp. 1631-1632.

(10) Saridis,G.N. and Stein,G.:"Stochastic approximation algorithms for linear discrete-time system identification", IEEE Trans. on Automatic Control, vol. ac-13, October 1968, pp. 515-523.

(11) Naguma,J. and Noda,A.:"A Learning Method for System Identification", IEEE Trans. on Automatic Control, vol. ac-12, pp. 282-287, 1967.

(12) Smith, F.W. :"System laplace-transform estimation from sampled data", IEEE Trans. on Automatic Control, vol. ac-13, pp. 37-44.

(13) Sinha,N.K.: "Estimation of transfer function of continuous system from sampled data", Proc. IEE, vol. 119, May 1972, pp. 612-614.

(14) Bereznai, G.T. and Sinha, N.K.: "Adaptive Control of Nuclear Reactors Using a Digital Computer", IEEE Trans. on Nuclear Science, vol. NS-18 August 1971, pp. 387-397.

(15) Bereznai,G.T. and Sinha,N.K.: "Application of optimum low-order models to the adaptive control of nuclear reactors", Proceedings of 5th IFAC Congress, Paris, 1972.

(16) Sinha,N.K. and Griscik,M.P.: "A Stochastic Approximation Method", IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-1, October 1971, pp. 338-344.

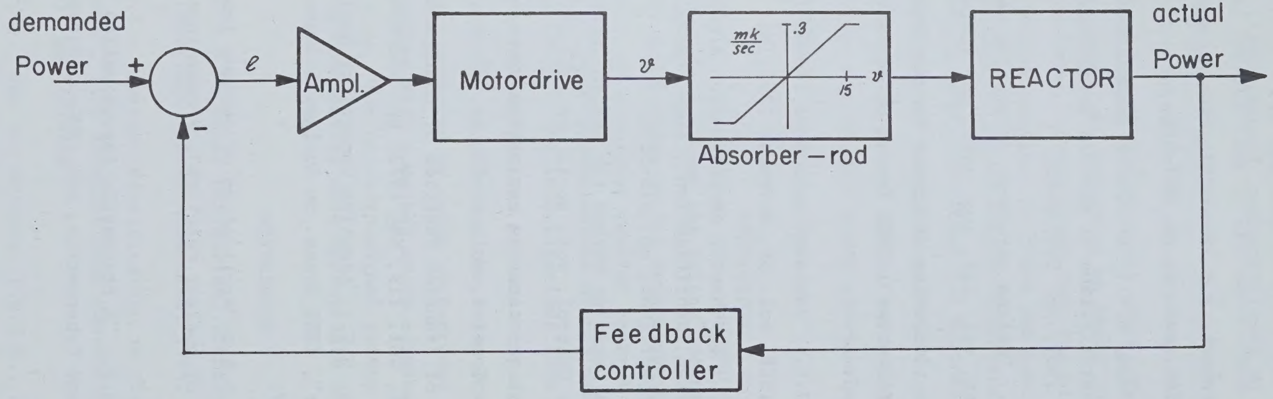

demanded
Power
+
-
ℓ
Ampl.
Motordrive
ϑ
$\frac{mk}{sec}$
.3
15
ϑ
Absorber – rod
REACTOR
actual
Power
Feedback
controller

FIG. I Simplified block diagram of Reactor and control system.

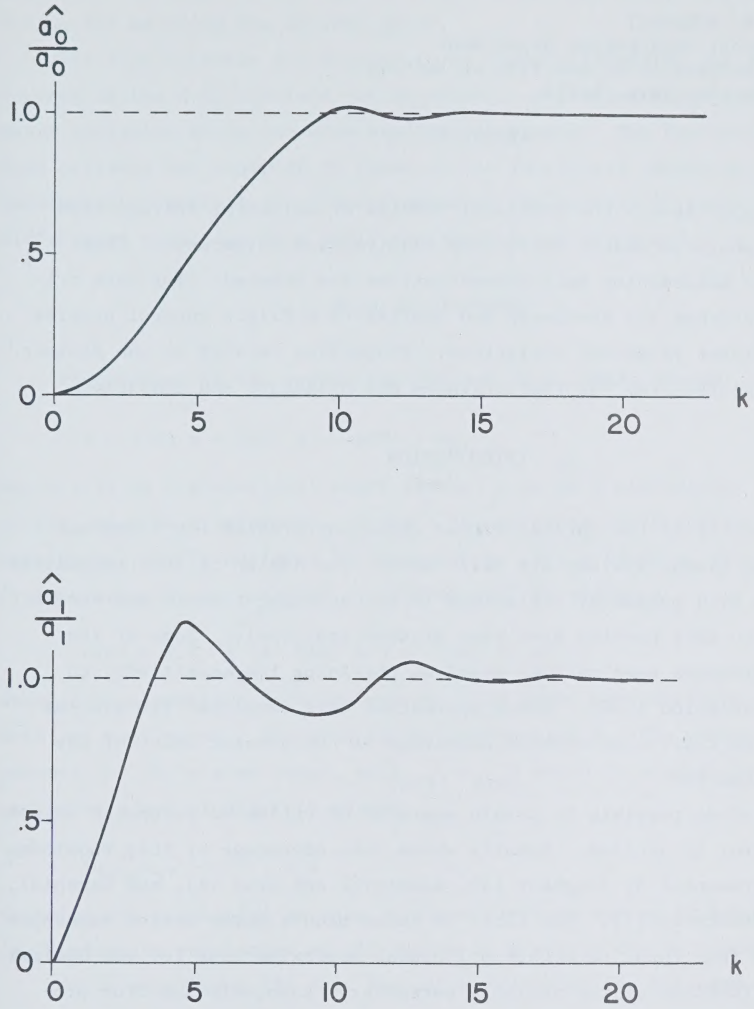


FIG. 2 Variation of the estimates of a_0 and a_1 with number of samples for 10% noise.

APPLICATION OF FOUR METHODS FOR APPROXIMATING OPTIMAL FEEDBACK GAINS

JAMES H. BURGHART
Electrical Engineering Department
State University of New York at Buffalo
Buffalo, New York 14214

ABSTRACT

This paper treats the practical problem of adjusting optimal feedback gains where parameter variations require such adjustment. Four criteria for determining gain approximations are treated. For each criterion, algorithms are developed and applied to a flight control problem with significant parameter variations. Properties related to the dynamic control resulting from the four criteria are discussed and compared.

I. INTRODUCTION

The results of the optimal linear regulator problem for fixed and time-varying linear systems are well-known. The design of such regulators for systems with parameter variations is not straight-forward and several approaches to this problem have been studied previously. Some of the earlier approaches used methods based on obtaining low sensitivity to parameter variation [1-4]. These approaches were developed for systems which must be controlled without knowledge of the present value of the variable parameter.

It is often possible to obtain measured or estimated parameter values before control is applied. Results which take advantage of this knowledge have been presented by Burghart [5], Kokotovic and Cruz [6], and Jamshidi, D'Ans and Kokotovic [7]. The first of these used a power series expansion of the cost functional to obtain polynomial approximations for the optimal gains as a function of the variable parameter. Kokotovic and Cruz presented a theorem dealing with properties of such expansions while Jamshidi et.al. developed a method based on numerical integration of a matrix differential equation to generate optimal gains when the parameter value becomes available.

This paper is concerned with investigating the effects of various criteria for obtaining polynomial approximations for the optimal gains. These approximations are to be used for simple on-line implementations in which gains are adjusted when new values of the variable parameter become available (see Fig. 1). Four different criteria are considered. Three of these are based on values of the cost function for various conditions of parameter variation and the fourth is a conventional curve-fitting criterion for matching the optimal gains.

The four criteria are applied to an example involving the longitudinal control of the X-15 aircraft during reentry, a problem with extensive parameter variation which has been studied previously. The results for all four criteria are superior to those of the functional expansion method. Conclusions regarding the effectiveness of the various criteria and the differences resulting from their application are discussed.

II. PROBLEM STATEMENT

The problem is the regulating control of a linear system

$$\dot{x} = A(\alpha)\, x + B(\alpha)\, u, \quad x(0) = x_0 \tag{1}$$

where x is an n-dimensional state vector, u is an r-dimensional control vector and α is a parameter. A and B are matrices which are assumed to be continuously differentiable functions of α. For a fixed α, desirable dynamic behavior is achieved by finding u(t) to minimize

$$J(\alpha) = 1/2 \int_0^\infty (x'\, Q(\alpha)\, x + u'\, R(\alpha)\, u)\, dt \tag{2}$$

where R is a positive definite matrix, Q is a pos. semi-definite matrix and both are continuously differentiable functions of α. The parameter α is assumed to lie in some range, $\alpha \varepsilon [\alpha_{min}, \alpha_{max}]$ which is adequately represented by a set of discrete values

$$\{\alpha_1, \alpha_2, \text{-----}, \alpha_N\}$$

Satisfactory regulation at each of these discrete values is assumed to imply adequate regulation at all values of α within the pertinent range.

It is well-known [8] that the optimum control for the case $\alpha = \alpha_i$ is

$$u = -R^{-1}\, B'(\alpha_i)\, K(\alpha_i)\, x = -G^*(\alpha_i)x \qquad (i = 1,2,\text{---},N) \tag{3}$$

where $K(\alpha_i)$ is the steady state solution to the matrix Riccati equation

$$K(\alpha_i)\, A(\alpha_i) + A'(\alpha_i)\, K(\alpha_i) - K(\alpha_i)\, B(\alpha_i)\, R^{-1}(\alpha_i)\, B'(\alpha_i)\, K(\alpha_i) + Q(\alpha_i) = 0 \tag{4}$$

and $G^*(\alpha_i)$ is the optimal gain matrix. Clearly, the gain matrix of (3) is a function of α_i and computing its value would normally require solving the steady-state Riccati equation each time the current value of α becomes available. Other possible approaches include storing $G^*(\alpha_i)$ for each α_i, numerically integrating an imbedding equation [7] which is derived from the Riccati equation or using approximating functions to obtain an approximate realization of $G^*(\alpha)$, $G(\alpha)$, where the actual control to be implemented is then

$$u = -G(\alpha)x \tag{5}$$

The approach considered here involves the gain approximating functions. Two previous approaches [5,7] to this problem treated a power series expansion of the cost functional in terms of powers of α. The present approach is based on direct minimization of certain criterion functions in order to find the coefficient matrices of the polynomial approximation

$$G(\alpha) = G_0 + G_1 \alpha + G_2 \alpha^2 + --- \tag{6}$$

Such an approach is advantageous because it specifically considers results for the N discrete cases of parameter variation and because it does not depend on functional expansions to be accurate at values of α far away from the point of expansion.

In this paper, four different criteria are considered for determining G_0, G_1, G_2 --- of (6). The object is to investigate the effect of these gain-approximating criteria on the control of a system with parameter variation and to compare the complexity of the computational algorithms of the minimizations associated with them.

III. CRITERIA FOR GAIN APPROXIMATION

Three of the four criteria used for deriving approximations for the gains as functions of α use values of the cost J (of (2)) evaluated at the N discrete values of α. The fourth criterion is the conventional least squares criterion for fitting the approximate gains to the optimal gains without regard to J.

The value of J (of (2)) which results when α takes on the value α_i is denoted $J_i (i = 1,2,---,N)$. It has been shown by Meditch [9] that J_i may be calculated using the unique solution to the linear matrix differential equation

$$\frac{dP_i}{dt} + (A(\alpha_i) - B(\alpha_i)G)'P_i + P_i(A(\alpha_i) - B(\alpha_i)G) + G'\,R(\alpha_i)G + Q(\alpha_i) = 0 \tag{7}$$

with boundary condition $P_i(\infty) = 0$ where

$$J_i = 1/2\ x_0'\ P_i(0)\ x_0. \tag{8}$$

$P_i(t)$ is called the cost matrix for the case $\alpha = \alpha_i$. Since the time interval for optimization is infinite, $P_i = P_i(0)$ is the steady state solution to (7) which satisfies the linear algebraic matrix equation

$$(A(\alpha_i) - B(\alpha_i)G)'P_i + P_i(A(\alpha_i) - B(\alpha_i)G) + G'\,R(\alpha_i)G + Q(\alpha_i) = 0 \qquad (i = 1,2,---,N) \tag{9}$$

Chen and Shieh [10] have described a simple method for solving equations of this type by treating a system of n(n+1)/2 linear algebraic equations. With the assumption of a stable controller, $(A(\alpha_i) - B(\alpha_i)\ G(\alpha_i))$ is a matrix whose eigenvalues have negative real parts (for all i) and the solutions to (9) are unique. Three of the criteria for optimizing the gain approximations involve costs associated with α_i which are to be calculated in this manner from P_i.

The first criterion for determining the coefficient matrices of the gain approximation (6) is the Average Cost Criterion

$$C_{AV} = \frac{1}{N} \sum_{i=1}^{N} J_i \tag{10}$$

which includes the value of the cost function for each α_i. The minimization of this criterion tends to produce a gain approximation which yields low values of J over the entire applicable range of α. A fixed x_0 must be specified.

The second criterion is the Relative Error Cost

$$C_{RE} = \sum_{i=1}^{N} \frac{J_i - J_{i_{opt}}}{J_{i_{opt}}} \tag{11}$$

which uses the optimal value of the cost functional for each α_i

$$J_{i_{opt}} = 1/2\ x_0'\ K(\alpha_i)x_0 \qquad (i = 1,2,---,N) \tag{12}$$

The minimization of this criterion tends to produce equal relative errors over the range of α whereas the Average Cost Criterion tends to favor cases of parameter variation which yield high values of J_i. Here again,

x_0 is fixed and must be specified.

The third criterion is the Trace Cost Criterion

$$C_{TR} = \sum_{i=1}^{N} \text{trace}\ (P_i) \tag{13}$$

which has been used in other applications [11,12] where x_0 cannot be specified. The trace operator has the effect of averaging initial states so that the value of x_0 does not affect the value of the criterion function.

The fourth criterion is a conventional least-squares fit (with uniform weighting)

$$C_{LS}^{(i,j)} = \sum_{i=1}^{N} (g_{ij}(\alpha_k) - g_{ij}^*(\alpha_k))^2 \qquad \begin{pmatrix} i = 1,2,\text{---},r \\ j = 1,2,\text{---},n \end{pmatrix} \tag{14}$$

which is applied to each element of the gain matrix, where $G(\alpha_k) = \left\{g_{ij}(\alpha_k)\right\}$ and $G^*(\alpha_k) = \left\{g^*_{ij}(\alpha_k)\right\}$. The least-squares fit tends to make the approximate gains "close" to the optimal gains without regard to the cost functional.

All four criteria are functions of the gain approximation matrices G_0, G_1,---. Necessary conditions for these matrices to minimize the various criteria and computational algorithms for computing them are now discussed.

IV. NECESSARY CONDITIONS AND COMPUTATIONAL ALGORITHMS

Necessary conditions for minimization are obtained by setting to zero the first partial derivative of each criterion function with respect to each element of G_0, G_1, G_2,--- from (6). Except for the least-squares criterion, this requires calculating partial derivatives of the elements of P_i (all i) with respect to elements of G_0, G_1, etc.

Let g_t denote a typical element from one of the matrices $\{G_0, G_1, G_2\text{---}\}$. Then using (8) and considering the criteria of (10), (11) and (13), the necessary conditions for optimal g_t are

$$\frac{\partial C_{AV}}{\partial g_t} = \frac{1}{2N} \sum_{i=1}^{N} x_0' \frac{\partial P_i}{\partial g_t} x_0 = 0 \tag{15}$$

$$\frac{\partial C_{RE}}{\partial g_t} = 1/2 \sum_{i=1}^{N} x_0' \frac{\partial P_i}{\partial g_t} x_0 \Big/ J_{i\,opt} = 0 \tag{16}$$

and

$$\frac{\partial C_{TR}}{\partial g_t} = \sum_{i=1}^{N} \text{trace } (\frac{\partial P_i}{\partial g_t}) = 0 \tag{17}$$

Each of these requires $\partial P_i/\partial g_t$ which may be calculated from (9). It follows from the theorems on differentiability of solutions with respect to parameters of differential equations that P_i is a differentiable function of g_t which satisfies (for i = 1,2,---,N)

$$(A(\alpha_i)-B(\alpha_i) \frac{\partial G}{\partial g_t})'P_i + (A(\alpha_i)-B(\alpha_i)G) \frac{\partial P_i}{\partial g_t} + \frac{\partial P_i}{\partial g_t} (A(\alpha_i)-B(\alpha_i)G)$$
$$+ P_i(A(\alpha_i)-B(\alpha_i) \frac{\partial G}{\partial g_t}) + (\frac{\partial G}{\partial g_t})' R(\alpha_i)G + G' R(\alpha_i) \frac{\partial G}{\partial g_t} = 0 \tag{18}$$

After solving (9) for P_i, $\partial P_i/\partial g_t$ is obtained from (18) which represents a set of linear algebraic equations in the elements of $\partial P_i/\partial g_t$. Here again the solution is unique if G stabilizes the system.

The necessary condition for minimizing the least-squares criterion is well-known and therefore not repeated here. A simple matrix inversion is required for the coefficients of each element of $G(\alpha)$.

The gradient conditions of (15), (16) and (17) suggest computational algorithms based on a gradient method. Such algorithms were developed using the Fletcher-Powell-Davidon method [13] for unconstrained parameter optimization problems. A computer program was developed to optimize the gain approximations based on these criteria. Gradient functions were evaluated using equations of the form of (18).

V. FLIGHT CONTROL EXAMPLE

The pitch-axis control of the X-15 aircraft during reentry was chosen as an example for evaluating the four criteria for approximating optimal gains. This example is particularly interesting because of the great range of parameter variation: the dynamic pressure, q, varies in the range [22.1, 1043]. In addition, the problem has been treated previously in the literature [5,7,14].

The linearized incremental longitudinal equations of motion with $x(t) = [\alpha \ \ \dot{\theta} \ \ \delta_e]$ and $u(t) = \delta_c$ were taken to be the same as previously published[5,7]. Here α is the incremental angle of attack, θ is the pitch rate, δ_e is the incremental elevator angle and δ_c is the control input to the actuator. Q and R had been determined originally [5] to provide acceptable regulation for any constant q in the range [22.1, 1043]. All results were evaluated using the initial condition $x(0) = [1 \ \ 0 \ \ 0]'$.

The minimization algorithms of the last section were used to optimize the coefficients of the gain approximations for each of the four criteria. It was desired to investigate how close these approximations approached the optimal gains and how well the optimal cost as a function of q was approximated. Both linear and quadratic gain approximations were considered but only the results for the quadratic approximations are shown since the linear approximations proved inadequate. It is important to note that even a quadratic approximation proved inadequate for the entire range of q with the previous functional expansion approach [5].

The optimal cost and the costs resulting from the various approximations are given in Figs. 2 and 3 for the range of parameter variation. At low values of q (high altitude, very lightly damped case) the optimal cost is best approximated by the results from the Average Cost Criterion. This is due to the fact that the cases corresponding to this range produce very high values of J(q) (compared to that from q in a higher range) and dominate contributions to C_{AV} (10). The Relative Error Criterion yielded an approximation which provided more uniform cost matching due to the normalization by the optimal cost. The Trace Criterion produced the poorest cost matching, probably because it did not take specific account of the initial state as did the two previous criteria. One of the most interesting aspects of these results is the excellent results from the Least Squares Criterion whose cost function is second only to that of the Average Cost Criterion. This is particularly important because the computational effort for curve fitting is much less than that for the other criteria since its algorithm is not iterative like the others.

The optimal and approximate gain functions for the various methods are shown in Figs. 4, 5, 6. It is interesting that gains which deviate so much from optimal can produce cost functions so close to optimal. The approximate gains from the Least Squares Criterion naturally are the best approximations to the optimal gains since they were calculated with this objective.

All of the approximate gain functions shown here provided stable control action over the prescribed range of q.

VI. CONCLUSIONS

The direct methods for gain approximation all were capable of giving satisfactory regulating control action for the example. All were far superior to the earlier indirect functional expansion method, although this method was computationally easier than the iterative methods based on gradient searches.

As a general rule, closeness of the approximate gains to the optimal gains does not necessarily imply closeness of the approximate cost to the optimal cost, and vice versa.

The Trace Criterion appears to be the poorest choice when x_0 is known but is one of the few possibilities when x_0 is unknown. The Average Cost Criterion is the best choice for matching the optimal cost function but the Relative Error Criterion provides a more uniform cost matching. The Least Squares Criterion is computationally simplest and provided very good results for the example.

In choosing between the Average Cost Criterion and the Relative Error Criterion, it is important to note that the Average Cost Criterion favors cases of underdamped dynamics, where the cost function takes on high values, whereas the Relative Error Criterion favors overdamped cases where the cost function is small.

REFERENCES

[1] J.A. Cadzow, "Optimal control of a system subject to parameter variation," Preprints, 1966 JACC (Seattle, Wash.), pp. 807-810.

[2] B.S. Morgan, Jr., "Sensitivity analysis and synthesis of multi-variable systems," IEEE Trans. Automatic Control, vol. AC-11, pp. 506-512, July 1966.

[3] W.R. Perkins and J.B. Cruz, Jr., "The parameter variation problem in state feedback control systems," Trans. ASME, J. Basic Engrg., ser. D, vol. 87, pp. 120-124, March 1965.

[4] R. Rohrer and M. Sobral, Jr., "Sensitivity considerations in optimal system design," IEEE Trans. Automatic Control, vol. AC-10, pp. 43-48, January 1965.

[5] J.H. Burghart, "Suboptimal linear regulators for systems subject to parameter variations," IEEE Trans. Automatic Control, vol. AC-14, pp. 285-289, June 1969.

[6] P.V. Kokotovic and J.B. Cruz, Jr., "An approximation theorem for linear optimal regulators," Journal of Mathematical Analysis and Applications, vol. 27, pp. 289-292, August 1969.

[7] M. Jamshidi, G.C. D'Ans and P. Kokotovic, "Application of a parameter-imbedded Riccati equation," IEEE Trans. Automatic Control, vol. AC-15, December 1970.

[8] M. Athans and P. Falb, "Optimal Control," Ch. 9, McGraw-Hill, New York, 1966.

[9] J.S. Meditch, "A class of suboptimal linear control," IEEE Trans. Automatic Control, vol. AC-11, pp. 433-439, July 1966.

[10] C.F. Chen and L.S. Shieh, "A note on expanding $PA + A^TP = -Q$," IEEE Trans. Automatic Control, vol. AC-13, p. 756, December 1968.

[11] J.H. Burghart, "Constant gain feedback controller for systems with uncertain parameters," ASME Paper No. 69-WA/Aut-13, presented at ASME Winter Annual Meeting, November 1969.

[12] D. Kleinman, T. Fortmann and M. Athans, "On the design of linear systems with piecewise-constant feedback gains," IEEE Trans. Automatic Control, vol. AC-13, pp. 354-361, August 1968.

[13] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization," Computer Journal, vol. 6, pp. 163-168, 1963.

[14] E. Kreindler, "Closed-loop sensitivity reduction of linear optimal control systems," IEEE Trans. Automatic Control, vol. AC-13, pp. 254-262, June 1968.

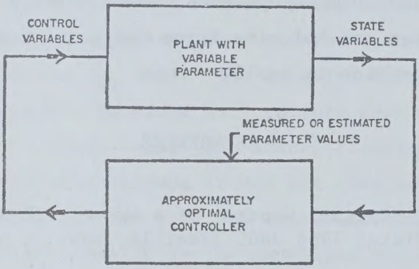

CONTROL VARIABLES
PLANT WITH VARIABLE PARAMETER
STATE VARIABLES
MEASURED OR ESTIMATED PARAMETER VALUES
APPROXIMATELY OPTIMAL CONTROLLER

FIG. 1. CONTROL CONFIGURATION

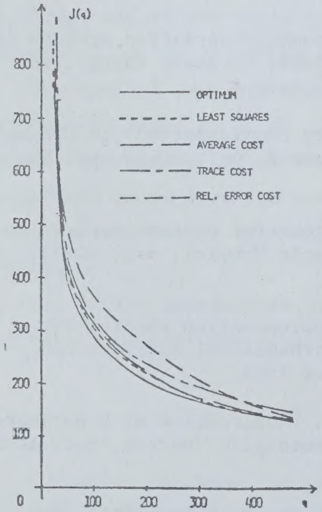


FIG. 2. COST CHARAC. (LOW q)

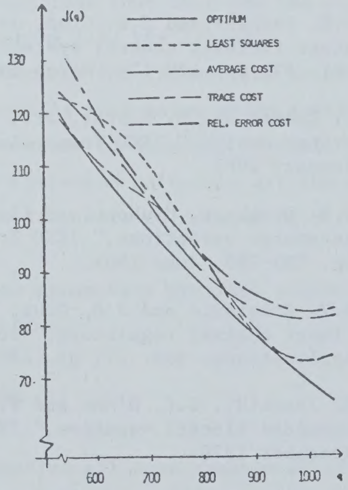


FIG. 3. COST CHARAC. (HIGH q)

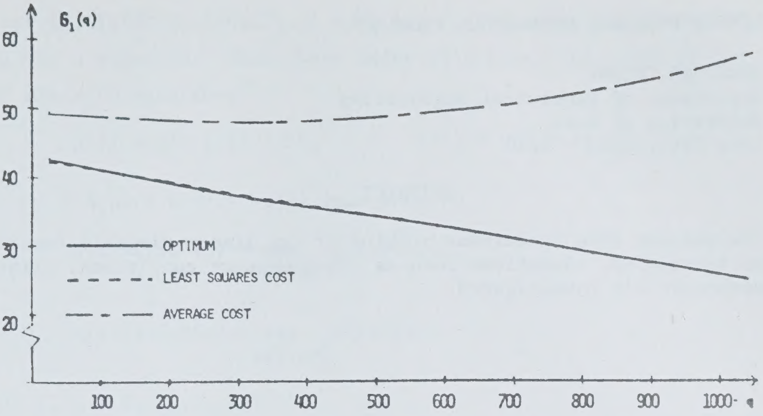


FIG. 4. OPTIMAL AND APPROXIMATE G_1

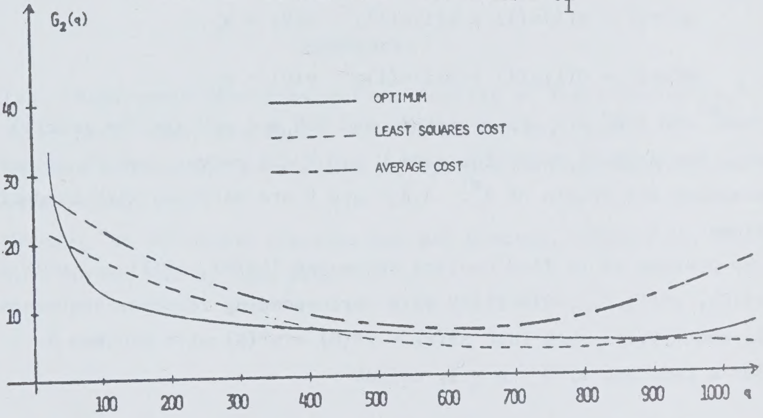


FIG. 5. OPTIMAL AND APPROXIMATE G_2

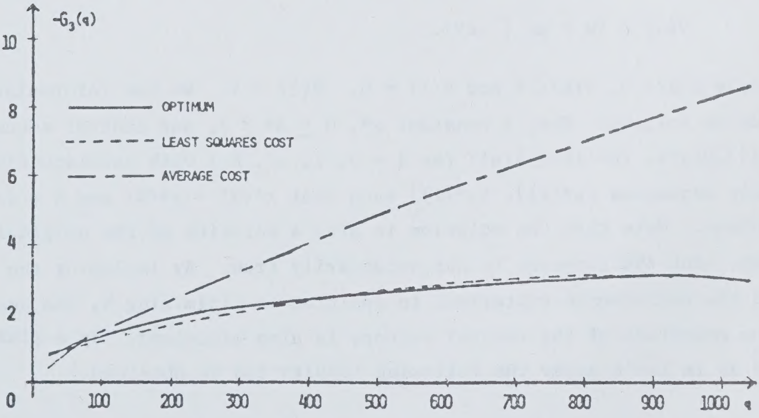


FIG. 6. OPTIMAL AND APPROXIMATE G_3

ON A LINEAR DISCRETE RENDEZVOUS PROBLEM*

DONG HAK CHYUNG
Department of Electrical Engineering
University of Iowa
Iowa City, Iowa 52240

ABSTRACT

The minimum time rendezvous problem of two linear discrete controlled systems is studied. Questions such as the necessary conditions, existence and uniqueness are investigated.

Consider two linear discrete vector systems

$$x(i+1) = A(i)x(i) + B(i)u(i), \quad x(0) = x_0,$$

$$y(i+1) = C(i)y(i) + D(i)v(i), \quad y(0) = y_0$$

where $x \varepsilon R^n$ and $y \varepsilon R^n$ are state vectors and $u \varepsilon U$ and $v \varepsilon V$ are the control vectors. The control restraint sets U and V are convex compact subsets in R^m containing the origin of R^m. A,B,C and D are matrices with compatible dimensions.

The problem is to find control sequences $\{u^*(0), u^*(1),\dots,u^*(N-1)\} \subset U$ and $\{v^*(0), v^*(1),\dots,v^*(N-1)\} \subset V$ with corresponding response sequences $\{x^*(i)\}$ and $\{y^*(i)\}$ such that $x^*(N) = y^*(N) = y^*(N)$ with minimum N.

For a constant a, $0 \leq a \leq 1$, define

$$U(a) = \{w = au \mid u \varepsilon U\}$$

$$V(a) = \{w = av \mid v \varepsilon V\}.$$

Obviously $U(a) \subset U$, $V(a) \subset V$ and $U(1) = U$, $V(1) = V$. We now reformulate our problem as follows: Find a constant a^*, $0 \leq a^* \leq 1$, and control sequences $\{u^*(i)\} \subset U(a^*)$, $\{u^*(i)\} \subset V(a^*)$ for $i = 0, 1,\dots, N-1$ with corresponding response sequences $\{x^*(i)\}$, $\{y^*(i)\}$ such that $x^*(N) = y^*(N)$ and $N + (a^*/2)$ is minimum. Note that the solution is also a solution of the original problem. But the converse is not necessarily true. By including the term $a/2$ to the performance criterion, in addition to minimizing N, the needed maximum magnitude of the control vectors is also minimized. By a similar method as in Lee's paper the following results can be obtained.

---

*This research was supported by NSF Grant No. GK31628.

Let a^*, $\{u^*(i)\}$, $\{v^*(i)\}$, $i = 0, 1, \ldots, N-1$, with responses $\{x^*(i)\}$, $\{y^*(i)\}$ be a solution. Then there exist solutions $p(i)$, $q(i)$ of the vector discrete equations

$$p(i) = p(i+1)\,A(i), \qquad q(i) = q(i+1)\,C(i)$$

for $0 \leq i \leq N$ with $p(N) = -q(N)$ such that

$$p(i)B(i)u^*(i) = \max_{u \in U(a^*)} p(i)B(i)u$$

$$q(i)D(i)v^*(i) = \max_{v \in V(a^*)} q(i)D(i)v$$

for all i, $i = 0, 1, \ldots, N-1$.

The existence and uniqueness of a solution is also investigated.

REFERENCES

E.B. Lee, "Recurrence Equations and the Control of Their Evolution," J. of Math. Anal. and Appl., Vol. 7, No. 1, Aug. 1963.

E.B. Lee and L. Markus, "Foundations of Optimal Control Theory," New York, Wiley, 1966.

D.H. Chyung, "On Nonunique Discrete Optimal Control," IEEE T-AC, Vol. AC-14, No. 3, June 1969.

ON OPTIMAL FEEDBACK CONTROL OF STOCHASTIC SYSTEMS AND CERTAIN ASSOCIATED MATHEMATICAL PROBLEMS.

N.U. AHMED
Department of Electrical Engineering
Ottawa University
Ottawa, Ontario, Canada.

INTRODUCTION

Consider the stochastic differential system S: $d\xi = F(t, \xi(t), u(t,\xi(t))dt + G(t, \xi(t), u(t, \xi(t)))dw(t)$, $t \geq o$ with $\xi(o) = \xi_o$ - P_o initial probability measure and $u \in C_o$ - the space of bounded measurable functions on $\{t \geq o\} \times R^n$ to $U \subset R^m$, U compact; F and G are measurable in t and continuous in the rest of the variables.

The problem is : find a control $u^* \in C_o$ that minimizes the cost functional J given by

$$J(u) = \mathcal{E}\{ \int_o^T f_o(t, \xi(t), u(t, \xi(t)))\, dt + \varphi(\xi(T)) \} .$$

Subject to the constraint that ξ is the solution of the stochastic differential equation corresponding to the controls $u \in C_o$. This problem is called P_1.

2 FIXED TIME PROBLEMS

Theorem 1.

Consider the system S with C_o as the class of admissible controls. Suppose (i) $\varphi : R^n \to R_o$ is continuous and (ii) $f_o : I \times R^n \times U \to R_o$ is bounded measurable on I for each $(x,u) \in R^n \times U$ and continuous on $R^n \times U$ for almost all $t \in I$. Then the problem P_1 is equivalent to the problem P_1' : $-\frac{\partial v}{\partial t} = A^u v + f_o(t,x,u(t,x)), (t, x) \in [o, T) \times R^n$; $v(T,x) = \varphi(x)$, $x \in R^n$, $u \in C_o$; $J(u) = \int_{R^n} v(o,x)\, P_o(dx) = \min_{u \in C_o}$.

where $A^u \psi = \sum_{i=1}^n F_i \frac{\partial \psi}{\partial x^i} + \sum_{i,j=1}^n a_{ij} \frac{\partial^2 \psi}{\partial x^i \partial x^j}$, $a_{ij} = \frac{1}{2}(GG')_{ij}$.

Corollary 2

The problem P_2: $d\xi = F^u dt + G^u dw$, $t \in [o, T]$, $\xi(o) = \xi_o$,
$J(u) = \mathcal{E}\{\int_o^T f_o(\theta, \xi(\theta), u(\theta, \xi(\theta)))\, d\theta\} = \min_{u \in C_o}$ is equivalent to the problem P_2' : $-\frac{\partial v}{\partial t} = A^u v + f_o^u$, $(t, x) \in [o, T) \times R^n$; $v(T, x) = o$, $x \in R^n$; $J(u) = \int_{R^n} v(o, x)\, P_o(dx) = \min_{u \in C_o}$.

Corollary 3

The problem P_3: $d\xi = F^u dt + G^u dw$, $t \in [o, T]$; $\xi(o) = \xi_o\ (P_o)$; $J(u) = \mathcal{E}\{\varphi(\xi(T))\} = \min_{u \in C_o}$ is equivalent to the problem P_3' : $-\frac{\partial v}{\partial t} = A^u v$ $(t,x) \in [o, T) \times R^n$; $v(T,x) = \varphi(x)$, $x \in R^n$; $J(u) = \int_{R^n} v(o,x)P_o(dx) = \min_{u \in C_o}$.

In certain problems of economics and pollution control it is required to choose a control that forces the system trajectory to spend maximum time in a desirable region of the state space. This is solved in the following:

Theorem 4.

The problem P_4: $d\xi = F^u dt + G^u dw$, $t \in [o,T]$; $\xi(o) = \xi_o(P_o)$; $J(u) = \mathcal{E}\{\nu(\ell(\xi^u))\} = \max\limits_{u \in C_o}$ where ν is the Lebesgue measure and $\ell(\xi^u) = \{t \in [o,T] : \xi^u(t) \in B,\ B \subset R^n \text{ (compact)}\}$, is equivalent to the problem P_4' : $-\frac{\partial v}{\partial t} = A^u v + C_B(x)$, $t \in [o,T)$, $x \in R^n$; $v(T,x) = o$, $x \in R^n$; $J(u) = \int_{R^n} v(o,x)\, P_o(dx) = \max\limits_{u \in C_o}$ where C_B is the characteristic function of the set B.

3 MARKOV TIME PROBLEMS

In case the terminal time is replaced by a Markov time, for example: $\tau_u = \text{Inf}\ \{t \geq o : \xi^u(t) \in \partial B\}$ where B is an open bounded set of R^n, then the Cauchy problems (Ths 1,4 cor 2,3) are replaced by corresponding Dirichlet problems.

Theorem 5.

The problem P_5: $d\xi = F^u dt + G^u dw$ $t \geq o$; $\xi(o) = \xi_o(P_o)$; $J(u) = = \mathcal{E}\{\int_o^{\tau_u} f_o{}^u\, d\theta + \varphi(\xi(\tau_u))\} = \min\limits_{u \in C_o}$ is equivalent to the problem

$$P_5' : \quad -\frac{\partial v}{\partial t} = A^u v + f_o^u,\ (t,x) \in [o,T) \times B;$$

$$v(t,x)\big|_{x \in \partial B} = \varphi(x),\ t \in (o,T],\ o < T < \infty;\quad J(u) = \int_B v(o,x)\, P_o(dx) = \min_{u \in C_o}.$$

4 EXISTENCE OF OPTIMAL CONTROLS

The problems of existence of optimal controls were studied in Fleming (1966) and Ahmed (1972) under very general conditions. For practical application, however, the question of existence of optimal feedback controls from the class C_o appears to be more interesting. Some preliminary results are presented below

Lemma 6 :

Suppose f_o is continuous and convex on U for almost all $(t,x) \in I \times R^n$ and $f_o^u \in L_p(I, H_o)$, $1 < p < \infty$, $H_o = L_2(R^n, P_o)$, for each $u \in C_o$. Then the set $C_1 = \{f_o^u;\ u \in C_o\}$ is a convex and weakly sequentially compact subset of $L_p(I, H_o)$.

Theorem 7

Suppose the domain D of the operators $\{B^u \triangleq \frac{\partial}{\partial t} + A^u, u \in C_o\}$ is a weakly closed subset of $L_p(I, H_2)$, $H_2 = \{z : z \in H_o, \frac{\partial z}{\partial x^i} \in H_o, \frac{\partial^2 z}{\partial n^i \partial n^j} \in H_o\}$ $(B^u(v_n - v_o), y) \to o$ uniformly on C_o whenever $v_n \xrightarrow{w} v_o$; and $u_n \xrightarrow{w^*} u_o$ implies $(B^{u_n} - B^{u_o})v, y) \to o$ for every $v \in D$, $y \in L_q(I, H_o)$. Then there exists optimal controls for the problems P_i $(i = 1 \ldots 6)$.

REFERENCES

W.H. Fleming (1966), J. Math. Mech. 15,5, pp 777-794

N.U. Ahmed (1972), J. Inf and Control (to appear).

LOCAL BALANCE IN QUEUEING NETWORKS WITH INTERDEPENDENT SERVERS

MANI CHANDY
The University of Texas at Austin
Austin, Texas 78712

MODEL

Consider a queueing system with S servers and C different classes of customers.

Customers

The classes of customers are indexed 1,2,...,C with generic indices b and c. Associated with each class of customers is a unique set of modes; at any given time a given customer can be in one and only one mode and the customer's behavior is a function of the mode he is in. A given mode belongs to one and only one class of customers, i.e., a customer of one class cannot enter a mode of another class. Let the set of modes of all customers be indexed 1,2,...,N with generic indices q,M and m. When a customer is in mode m he requires the services of a particular server, say server s_m; when he finishes receiving service from s_m he may change his mode or leave the system. A customer cannot change his current mode until he finishes getting service in that mode. The service times for all customers in all modes are exponentially distributed and independent. Let the mean service time in mode m be 1/u(m). A customer changes his mode to mode m when he finishes service in mode q with probability p(q,m) independent of all other customers, servers and his past behavior. Suppose mode m can only be entered by customers of class b and let p(0,m) be the probability that a customer who finishes service in mode m leaves the system.

A given class of customers may be either open or closed. The number of customers in any closed class is always a constant and customers of this class neither enter nor leave the system and p(0,m) = p(m,0) = 0 for all modes m for this class of customers. Open class customers may enter or leave the system. Let r_b be the number of class b customers in the system at some instant of time. Then r_b is a constant for all closed classes and may vary with time for open classes. Customers of (open) class b enter the system in an independent Poisson process with rate $v_b(r_b)$ which is a function of r_b the number of class b customers in the system.

Considering only the instants at which a given customer changes modes (and not the time spent in each mode) the mode that a customer is in after the nth mode-transition n = 1,2,... forms a Markov process for all closed classes of customers. These processes are all assumed to be irreducible and single chain.

Servers

There are S servers. A given server serves customers at several stations; the server moves from station to station according to a discipline described later. At any given time a given server can serve only one customer at only one station. Each server has his own distinct set of stations, i.e., a given station can be served by one and only one server. The time taken for a server to switch from one station to another is zero.

Each mode can be served at one and only one station. A station can service customers in several modes, however, the mean service time for all modes serviced at a given station must be the same.

We shall assume for ease of exposition that the service discipline in which customers at a given station are served is first come first served. However, precisely the same results and analysis holds for the last come first served preemptive resume discipline. Similar analysis holds for any discipline which serves customers at a station in a sequence which is independent of the modes of all customers and in which a server is not idle if there is a customer waiting to be served.

If the server is at station i_1 and a new customer arrives in station i_2 of that server, then the server immediately switches to station i_2 and starts serving customers there. Service is preemptive-resume, i.e., a customer resumes service from the point of preemption. The server remains at a given station until either (a) a new customer arrives at some other station of that server or (b) a customer finishes service at that server. For ease of exposition we shall assume that the server is always at that station which had the latest entry. However, a similar analysis holds for the case where, when a customer finishes service at a given station, the next station to be served is chosen by any discipline where the station selected is independent of the modes of all customers. For ease of exposition, the stations are indexed 1,2,...,L such that stations of different servers have different indices; thus specifying the index of a station also specifies the unique server who may service that station.

Given our assumption for the discipline in which the server switches from station to station, the sequence of stations served forms a stack: when a customer enters a given station its index is "pushed down" the top of the stack; the server always serves that station on top of the stack and when a single customer completes service at that station, the station's index is "popped-off" the stack. (A single station may have several entries in the stack). Each station has a queue of customers in some specified sequence (FCFS or LCFS or .....).

Server-group

Each one of the S servers belongs to one and only one server-group. A server-group consists of one or more parallel servers. A server's service rate is a function of the total number of customers (in queues for servers) in the corresponding server group.

Let station i be served by server j who belongs to server-group k. Let $\underline{l}_i$ be the number of customers in station i, n_j be the total number of customers in (all the stations for) server j, and w_k be the number of customers in (all the stations in all the servers in) server-group k. The service rate at a given station is a function of three variables: The number of customers at that station, in that server and in the server-group. In this paper we shall limit attention to the special case where the service rate (when the server is at station i) is $f_i(\underline{l}_i) \cdot g_j(n_j) \cdot h_k(q_k)$ times the service rate when there is only one customer in that server-group, and that customer is in station i; $f_i(\ )$, $g_j(\ )$ and $h_k(\ )$ are arbitrary functions which can only take on positive values. For instance the service rate for a customer in mode m is $f_i(\underline{l}_i) \cdot g_j(n_j) \cdot h_k(w_k) \cdot u(m)$. We shall call $f_i(\ )$, $g_j(\ )$ and $h_k(\ )$ the rate functions for station i, server j and server group k respectively. The product $f_i(\underline{l}_i) \cdot g_j(n_j) \cdot h_k(w_k)$ will be called the service rate multiplier. A customer's service rate multiplier is the ratio of the service rate he is actually getting to the rate he would have got if he were alone in the server group. Apart from the interdependence of service rates the different servers in a group are independent. Thus the stations in the system are arranged in a two-level hierarchy: the set of all stations are partitioned into servers and the set of all servers are partitioned into groups. A server serves only one station at a time and moves from station to station; all the servers in a group may operate in parallel though their service rate may be a function of the number of customers in the group.

It is assumed that the Markov process defined by this system is irreducible and ergodic.

MODEL VARIATIONS

The model discussed in the previous section will be referred to as the basic model. Some variations of the basic model are described below; these variations easily reduce to the basic model.

Rational Service Distribution

Consider a system identical to the basic model except that the service time in a mode is any rational distribution, i.e., a distribution with a rational Laplace transform. Any rational distribution can be represented as a network of exponential stages [6,7] where the service time in any stage is an independent exponential random variable. Replacing each stage of each rational distribution by a separate mode, this variation reduces to the basic model.

Processor Sharing Service Discipline

Consider a server-group where each server in the group serves customers in one and only one mode; hence each server has one and only one station. Let the functions for the station, server and server-group be $f_i(\underline{1}_i) = 1$ all $\underline{1}_i \neq 0$, $g_j(n_j) = n_j$ all n_j, and $h_k(w_k) = 1/w_k$ for $w_k \neq 0$; the service rate multiplier is then n_j/w_k which is the fraction of customers in the service group who are in the queue for server j. This is equivalent to the case where each of the n_j customers in the queue for server j are being serviced in parallel by n_j servers each with service rate multiplier $1/w_k$ and hence the server group as a whole acts like a single server with a processor-sharing discipline.

If the rate function for the server group is $h_k(w_k) = 1$ for $w_k \neq 0$, the service rate multiplier becomes n_j and hence the server group as a whole acts like a set of an arbitrarily large number of servers so that all customers in the queue are served in parallel.

Last Come First Served Preemptive Resume Service (LCFSPR) Discipline

In the LCFSPR discipline customers who arrive later at a queue are served before customers who arrived earlier. If every station of a given server was serviced with the LCFSPR discipline then the server acts like a server without stations serving customers in a LCFSPR manner.

Networks of Servers

Consider the model analyzed by Jackson [8] which consists of a network of servers and only one class of customers. When a customer finishes service at server i he joins the queue for server j with probability p(i,j) independent of the current state of the system. All service times are assumed to be independent and exponentially distributed. This model can be handled as a variation of the basic model by establishing a one-to-one correspondence between a server in the network model and a mode in the basic model.

Now consider an extension of Jackson's model studied by Chandy, Keller, and Browne [9]. In this model some of the servers use the processor-sharing (PS) discipline, others the LCFSPR discipline and others the FCFS discipline. The service distribution for servers using PS or LCFSPR may be any rational distribution. The service distribution for servers using FCFS must be exponential. There are several classes of customers each with its own service distributions and branching probabilities. However, all customers using a given FCFS server must have the same mean service time. Each customer has a set of modes and changes modes in a Markovian manner; a customer's service distribution and branching probabilities are a function of the mode the customer is in. This network model is evidently a variation of the basic model since modes, rational distributions, PS discipline, LCFSPR discipline etc. can be obtained from the basic model.

Several other variations of the basic model have been studied [8 - 18] in the literature. However, not all aspects of the basic model have been analyzed.

THEORY

The state of the jth server at any given instant of time is defined to be the sequence $A_j = (m(j,1),m(j,2),\ldots,m(j,d),\ldots,m(j,n_j))$ where $m(j,d)$ is the mode of the dth customer awaiting service from the jth server $d = 1,\ldots,n_j$ and the dth customer is served before the d+1-th for $d = 1,2,\ldots,n_j - 1$, and n_j is the number of customers getting or awaiting service from server j. (The modes $m(j,d), d = 1,\ldots,n_j-1$, may be serviced at several stations of the jth server.)

The states of the system are the cartesian products of the states of the servers: we shall denote the system state as $T = (A_1,A_2,\ldots,A_j,\ldots,A_S)$ where A_j is the state of the j-th server and there are S servers in the system. The system state is said to be feasible if the total number of customers of each given closed class b is equal to the given number of customers r_b in that class.

It has been assumed that the Markov process defined by the system is irreducible and ergodic. When the system is in equilibrium the rate at which the system enters a state is equal to the rate at which the system leaves that state. The system can leave a state in one of several ways; the system leaves a state if any customer finishes service with any one of the S servers. It is evident that the event: "system leaves state T" can be partitioned into S subsets: "system leaves state T because a customer finishes service with the jth server," $j = 1,2,\ldots,S$. Of course some of the S subsets may be empty; for instance if there are no customers getting service from the jth server in state T then the system cannot leave state T because a customer finishes service with the jth server. Similarly the event "system enters state T" may be partitioned into S subsets: "system enters state T because a customer arrives for service with the jth server," $j = 1,2,\ldots,S$. Here again some subsets may be empty. We shall use the abbreviation "system leaves (T,j)" for the subset of events "system leaves T because a customer finishes service with the jth server;" the abbreviation system enters(T,j) will be used for the subset of events "system enters T because a customer arrived for service with the jth server".

It is evident that for the basic model the system can enter(T,j) if and only if the system can leave(T,j). The system is said to be in local balance if the rate at which the system enters(T,j) is equal to the rate at which the system leaves(T,j), for all states T and all servers j. It is obvious that for the system considered here local balance is a sufficient condition for equilibrium.

The Manner in which the System Can Leave(T,j')

Let server j' be at station i' when the system is in state T. Let customers in station i' be served on a FCFS basis. Let the mode of the last customer in the queue for station i' be m' and the mode for the first customer in the queue be m_1, when the system is in state T. Let the j'th server belong to the k'th server-group and let ℓ_i, n_j, w_k be the number of customers in station i, server j and server-group k respectively, when the system is in state T.

The only way in which the system can leave (T,j') is if the customer in mode m_1 who is currently getting service from the j'th server, when the system is in state T, finishes being serviced. The rate at which this happens is:

$$\text{RATE LEAVE}(T,j') = P(T).f_{i'}(\ell_{i'})\cdot g_{j'}(n_{j'})\cdot h_{k'}(w_{k'})\cdot u(m_1) \qquad (1)$$

where P(T) is the equilibrium probability that the system is in T.

The Manner in which the System can Enter(T,j')

Since the j'th server is at station i' in state T the last customer to have arrived for service with the j'th server when the system enters(T,j') must have entered station i'. Since the mode of the last customer in the queue at station i' is m' it follows that the last customer to have arrived for service with the j'th server when the system enters(T,j') was in mode m'. Hence the system can enter(T,j') only if a customer in mode m' (and no other mode but m') arrives for service with the j'th server thus causing the system to transit to state T. Note that $u(m') = u(m_1)$ since m' and m_1 are served at the same station. Hence

$$\text{RATE LEAVE}(T,j') = P(T)\cdot f_{i'}(\ell_{i'})\cdot g_{j'}(n_{j'})\cdot h_{k'}(w_{k'})\cdot u(m') \qquad (2)$$

The notation $A_J + M$ will be used for the current state of the Jth server, $J = 1,2,\ldots,S$, if (a) the Jth server is currently serving a customer in mode M and (b) if this customer finishes service with the Jth server before any new customer arrives the state of the Jth server changes to A_J. In otherwords, if A_J is defined by the sequence $(m^1,m^2,\ldots,m^E)$ then $A_J + M$ is defined by $(M,m^1,m^2,\ldots,m^E)$ where customers earlier in the sequence get serviced before customers later in the sequence.

Let $A_{j'}$ be the state of the j'th server when the system is in state T. Recollect that the j'th server is at station i' when the system is in state T. The queue of customers in station i' is $(m_1,x^1,x^2,\ldots,x^D,m')$ where the first customer in the queue is in mode m_1 and the last in mode m' and the intermediate customers are in arbitrary modes. Let $A_{j'}-m'$ refer to the state of the j'th server, which is identical to $A_{j'}$ in every way except that the last customer (in mode m') is removed from the queue at station i; the queue for station i is now $(m_1,x^1,\ldots,x^D)$. In otherwords the current state of the j'th server is $A_{j'}-m'$ if when a customer changes his mode from some mode m to m' (and enters station i' of server j') the state of the j'th server changes to $A_{j'}$.

The notation $T(J + M, j'-m')$ will be used for the state $(A_1,\ldots,A_{J-1}, A_J + M, A_{J+1},\ldots,A_{j'-1},A_{j'}-m',A_{j'+1},\ldots,A_S)$. Let the customer in mode M who is getting service from the Jth server when the system is in state $T(J + M, j'-m')$ be served at station I. Let server J belong to server-group K. Let X_m,R_b,L_i,N_j,W_k be the number of customers in mode m, class b, station i, server j and server-group k respectively, for all m,b,i,j,k when the system is in state $T(J+m,j'-m')$, and let x_m,r_b,ℓ_i,n_j and w_k be the corresponding values in state T. Then $R_b = r_b$ all b, and

$$X_m = \begin{cases} x_m+1 & \text{for } m = M \\ x_m-1 & \text{for } m = m' \\ x_m & \text{for all other } m \end{cases}$$

$$L_i = \begin{cases} \ell_i+1 & \text{for } i = I \\ \ell_i-1 & \text{for } i = i' \\ \ell_i & \text{for all other } i \end{cases} \qquad (4)$$

$$N_j = \begin{cases} n_j+1 & \text{for } j = J \\ n_j-1 & \text{for } j = j' \\ n_j & \text{for all other } j \end{cases} \qquad (5)$$

$$W_k = \begin{cases} w_k+1 & \text{for } k = K \\ w_k-1 & \text{for } k = k' \\ w_k & \text{for all other } k \end{cases} \qquad (6)$$

The system transits from state $T(J+M,j'-m')$ to state T if the customer in mode M who is getting service from server J when the system is in state $T(J+M,j'-m')$ finishes service and changes mode to m' (and enters station i' of server j'). This transition will be denoted by $T(J+M,j'-m') \rightarrow T$ and the rate at which it occurs is

$$\text{RATE}\{T(J+M,j'-m') \rightarrow T\} =$$
$$P(T(J+M,j'-m')).f_I(L_I)\cdot g_J(N_J)\cdot h_K(W_K)\cdot u(M)\cdot p(M,m') \qquad (7)$$

Note that if a customer can change his mode from M to m' then modes M and m' belong to the same class of customers.

Let $T(j'-m')$ denote the system state $(A_1,\ldots,A_{j'-1},A_{j'}-m',A_{j'+1},\ldots,A_S)$. Let customers in modes m' and M belong to class b'. Let $x'_m, r'_b, \ell'_i, n'_j, q'_k$ be the number of customers in mode m, class b, station i, server j and server-group k respectively for all m,b,i,j,k when the system is in state $T(j'-m')$. Then

$$x'_m = \begin{cases} x_m-1 & \text{for } m = m' \\ x_m & \text{otherwise} \end{cases} \qquad (8)$$

$$r'_b = \begin{cases} r_b-1 & \text{for } b = b' \\ r_b & \text{otherwise} \end{cases} \qquad (9)$$

$$\ell'_i = \begin{cases} \ell_i-1 & \text{for } i = i' \\ \ell_i & \text{otherwise} \end{cases} \qquad (10)$$

$$n'_j = \begin{cases} n_j-1 & \text{for } j = j' \\ n_j & \text{otherwise} \end{cases} \qquad (11)$$

$$w'_k = \begin{cases} w_k-1 & \text{for } k = k' \\ w_k & \text{otherwise} \end{cases} \qquad (12)$$

The system can transit from $T(j'-m')$ to T only if b' is an open class; if a customer in open class b' enters the system in mode m' when the system is in state $T(j'-m')$, then the system will transit to state T. The rate at which this happens is

$$\text{RATE}\{(T(j'-m') \rightarrow T\}$$
$$= P(T(j'-m'))\cdot v_{b'}(r_{b'})\cdot p(0,m') \qquad (13)$$

If b' is a closed class then $p(0,m') = 0$ and equation (13) still holds. For simplicity in expression we shall put $v_b(r_b) = 1$ for all closed classes b, and of course $p(0,m) = 0$ if m belongs to a closed class. Then all the equations are valid for open and closed classes.

The net rate at which the system enters (T,j') is then

$$\text{RATE ENTER}(T,j') =$$
$$\sum_{M=1}^{N} \text{RATE}\{T(J+M,j'-m') \rightarrow T\} + \text{RATE}\{T(j'-m') \rightarrow T\} \qquad (14)$$

The system is said to be in local balance if

$$\text{RATE ENTER}(T,j') = \text{RATE LEAVE}(T,j') \text{ all } T,j' \qquad (15)$$

Definitions

Let y(m), m = 0,1,...,N be any set of numbers defined by the following relations:

$$y(0) = 1,\ y(m) = \sum_{M=0}^{N} y(M)\cdot p(M,m),\ m=1,\ldots,N \tag{16}$$

Such numbers exist given our assumptions of irreducibility and ergodicity. The following functions are defined recursively:

$$\left.\begin{array}{ll} F_i(0) = 1,\ F_i(n) = F_i(n-1)\cdot f_i(n) & \text{all } i \\ G_j(0) = 1,\ G_j(n) = G_j(n-1)\cdot g_j(n) & \text{all } j \\ H_k(0) = 1,\ H_k(n) = H_k(n-1)\cdot h_k(n) & \text{all } k \\ V_b(0) = 1,\ V_b(n) = V_b(n-1)\cdot v_b(n-1) & \text{all } b \\ U_m(0) = 1,\ U_m(n) = U_m(n-1)\cdot u(m) = \{u(m)\}^n & \text{all } m \\ Y_m(0) = 1,\ Y_m(n) = Y_m(n-1)\cdot y(m) = \{y(m)\}^n & \text{all } m \end{array}\right\} \tag{17}$$

Let Q( ) be the functional form for the equilibrium state probabilities, i.e.,

$$P(T) = \begin{cases} Q(T) & \text{if T is a feasible state} \\ 0 & \text{if T is an infeasible state} \end{cases}$$

Lemma 1: The functional form for the equilibrium probability that the system is in state T is:

$$Q(T) = \frac{\text{NORM.} \prod_{m=1}^{N} Y_m(x_m) \prod_b V_b(r_b)}{\prod_{m=1}^{M} U_m(x_m) \prod_i F_i(\ell_i) \prod_j G_j(n_j) \prod_k H_k(w_k)} \tag{18}$$

where NORM is a normalizing constant.

Proof: We shall show that this functional form satisfies local balance. By definition,

$$Q(T(j'-m')) = \frac{\text{NORM} \prod_{m=1}^{N} Y_m(x'_m) \prod_b V_b(r'_b)}{\prod_{m=1}^{N} U_m(x'_m) \prod_i F_i(\ell'_i) \prod_j G_j(n'_j) \prod_k H_k(w'_k)} \tag{19}$$

where x'_m, r'_b, ℓ'_i, n'_j, w'_k are defined in equations (8) - (12).

Hence

$$\begin{aligned} Q(T(j'-m')) &= \frac{Q(T)\cdot u(m')\cdot f_i(\ell_{i'})\cdot g_{j'}(n_{j'})\cdot h_{k'}(w_{k'})}{y(m')\cdot v_{b'}(r'_{b'})} \\ &= \frac{Q(T(J+M,j'-m'))\cdot u(M)\cdot f_I(L_I)\cdot g_J(N_J)\cdot H_K(W_K)}{y(M)\cdot v_B(r_B)} \end{aligned} \tag{20}$$

where mode M belongs to class B.

Substituting equation (18) in equation (7) and using equation (20)

$$\begin{aligned} &\text{RATE } \{T(J+M,j'-m') \rightarrow T\} = \\ &= Q(T(j'-m'))\cdot v_B(r_B)\cdot y(M)\cdot p(M,m') \\ &= Q(T(j'-m'))\cdot v_{b'}(r_{b'})\cdot y(M)\cdot p(M,m') \end{aligned} \tag{21}$$

since if a customer can change his mode from M to m', i.e., if $p(M,m') \neq 0$, then modes M and m belong to the same class, or $B = b'$.

Substituting equation (18) in equation (13),

$$\text{RATE } \{T(j'-m') \rightarrow T\}$$

$$= Q(T(j'-m')) \cdot v_{b'}(r_{b'}) \cdot p(0,m') = Q(T(j'-m')) \cdot v_{b'}(r_{b'}) \cdot y(0) \cdot p(0,m') \quad (22)$$

since $y(0) = 1$ by assumption.

Substituting (21) and (22) in (14), RATE ENTER(T,j')

$$= Q(T(j'-m')) \cdot v_{b'}(r_{b'}) \sum_{M=0}^{N} y(M) \cdot p(M,m')$$

$$= Q(T(j'-m')) \cdot v_{b'}(r_{b'}) \cdot y(m') \quad (23)$$

Substituting (18) in (2), RATE LEAVE(T,j')

$$= Q(T(j'-m')) \cdot v_{b'}(r_{b'}) \cdot y(m') \quad (24)$$

From (23) and (24) it follows that local balance (15) holds, since the above argument is valid for all T and j'. This completes the proof of the lemma.

ACKNOWLEDGEMENT

This work was carried out under National Science Foundation Grant GJ-1084

The autor would like to thank F. Palacios-Gomez for his help. The paper was based, in part, on the work on "job typing" carried out by Palacios-Gomez [14]. [16] Baskett and Muntz analyzed "job typing" independently of the work carried out by Palacios-Gomez and by Chandy, Keller and Browne [10].

[1] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, Mc-Graw Hill, New York (1964).

[2] E. G. Coffman and L. Kleinrock, "Feedback Queueing Models for Time-Shared Systems", JACM 15,4, pp. 549-576 (Oct. 1968).

[3] L. Kleinrock and R. R. Muntz, "Multi-level Processor-Sharing Queueing Models for Time-Shared Systems", *Proc. Sixth International Teletraffic Congress*, Munich Germany, pp. 341/1-341/8 (Aug. 1970).

[4] L. Kleinrock, "Time-Shared Systems: A Theoretical Treatment", *JACM*, 14, pp. 242-261, (1967).

[5] L. Kleinrock, R. R. Muntz, E. Rodemich, "The Processor-Sharing Queueing Model for Time-Shared Bulk Arrivals", *Networks*, 1, pp. 1-13 (1971).

[6] D.R. Cox, "A Use of Complex Probabilities in the Theory of Stochastic Processes", *Proc*. *Camb*. *Phil*. *Soc*., 51, pp. 313-319 (1955).

[7] P. M. Morse, *Queues, Inventories and Maintenance*, John Wiley, New York, (1958).

[8] J. R. Jackson, "Jobshop-Like Queueing Systems," *Man*. *Sci*. 10, pp. 131-142, (1963).

[9] K. M. Chandy, "The Analysis and Solutions for General Queueing Networks", *Proc. Sixth Annual Princeton Conference on Information Sciences and Systems*, Princeton Univ., Princeton, N. J. (March 1972).

[10] K. M. Chandy, T. W. Keller and J. C. Browne, "Design Automation and Queueing Networks", *Proc. Ninth Annual Design Automation Conference*, 9, pp. 357-367, (June 1972).

[11] K. M. Chandy, "Queueing Models in Computer Systems", *Proc. First Texas Symposium on Computer Systems*, pp. I-4-1 - I-4-7, Univ. of Texas at Austin, (June 1972).

[12] K. M. Chandy, "Local Balance in Queueing Networks with Several Classes of Customers", SIAM Annual Meeting, Philadelphia, (June 1972).

[13] K. M. Chandy and J. Hogarth, "The Improvement in Throughput Obtained by Parallel Processing: A Queueing Analysis", Workshop on Parallel Computation Jointly Sponsored by IEEE Computer Society and ACM, (June 1972).

[14] F. Palacios-Gomez, "An Analytic Model of a Multiprogramming System Including a Job Mix", Computer Sciences Dept. Report TR-4, Univ. of Texas at Austin (June 1972).

[15] F. Baskett and F. Palacios-Gomez, "Processor-Sharing in a Central Server Queueing Model of a Multiprogramming System with Applications", *Proc. Sixth Annual Princeton Conference on Information Sciences and Systems*, Princeton, N. J., (March 1972).

[16] F. Baskett and R. R. Muntz, "Queueing Network Models with Different Classes of Customers", *Proc. COMPCON 72*, San Francisco (Sept. 1972).

A COMPUTER-AIDED ROUTING TECHNIQUE FOR APPLICATION IN MANUFACTURING

W. J. RYCHLEWSKI AND R. W. MCLAREN
Department of Electrical Engineering
University of Missouri-Columbia

ABSTRACT

The technique summarized here is intended to establish an orderly general procedure to be applied in the development of computer-aided routings for a wide range of manufacturing applications. Some advantages of this approach are presented and an example of the successful application of the technique is briefly discussed.

INTRODUCTION

A routing can be defined as a document establishing an ordered sequence of all manufacturing operations required to produce a finished product while containing the necessary data to insure the proper performance of each operation. Conventionally, the preparation of a routing is a complex, time-consuming effort requiring the cooperation of a team of skilled individuals. A complete routing is usually necessary in response to produce modification orders that may require only minor design changes such as: changes in machine settings which determine finished dimensions, or in the order of operations to produce a special effect, or in the addition of extra operations to meet a customer request. By developing a computer-aided routing procedure, a detailed routing can be completed from a definitive set of finished product specifications, while the total time required to generate the routing is reduced to a matter of seconds.

SYSTEMATIC APPROACH

A systematic approach to developing a computer-aided routing follows:

A. Product definition and classification: Prepare a description of the finished product including all proposed variations. Determine the bases for dividing the product into classes. Prepare standards which clearly define an acceptable finished product. List major product components including information on the final assembly.

B. Decision criteria enumeration and reduction: Determine the decisions required to prepare a manual routing and enumerate sources of pertinent data. Eliminate data source duplication and develop new sources as needed. Attempt to isolate each decision, the associated decision criteria, and the data sources required.

C. Subdivision into specific operations: Subdivide the manufacturing process into major production operations which can be chained together to form a complete routing.

D. Formulation of a decision model: In order to establish the desired sequence of operations, an algorithm or model must be formulated utilizing the decision criteria. Often, a flowchart identifying decision points, possible operations associated with each decision point, and criteria for choosing the appropriate operation at each point will provide the requisite information to implement the model.

E. Selection of operational parameters: Mathematically derive as many of the operational parameters and machine settings as possible from specifications required of the finished assembly. This should reduce the number of inputs and eliminate many of the tables used in manual routing procedures. Much of the software maintenance required in computer-aided routing systems is directed toward file maintenance.

F. Software: Software considerations include speed, accuracy, storage minimization, program and data base maintainability, formatting, etc.

G. Hardware: Many routing systems are developed to be used on existing hardware. However, some new hardware may need to be acquired.
H. Testing: The logic, hardware, software, information acquisition system, report format and the like must be tested and evaluated.
I. Review, reprogram, and retest: Review of the test results by experts in production, software systems, and related fields will lead to changes and improvements in the system. After each change, reprogramming and retesting are required, followed by another review.
J. System implementation: When the system becomes operational, retraining of personnel may be required if the routing format has changed.
K. Continuous evaluation and periodic review: As new requirements arise, modifications and appropriate corrections should be made.

ADVANTAGES

Some advantages of computer-aided routing include the following:

1. To firmly establish a number of basic operations from which all products of a given type can be produced.
2. To establish well-defined routing standards for each basic operation and to apply these standards uniformly to all items produced.
3. To reduce inconsistency in product quality attributed to manual routings prepared by different individuals.
4. To eliminate calculation errors made by individuals.
5. To simplify procedures so that routings can be prepared by individuals unfamiliar with the processes involved.
6. To free skilled personnel for further product development.
7. To reduce the time interval from reception of the order to completion of the routing.
8. To provide a basis for further refinements in production operations through evaluation and improvement of present practices.

An automotive piston ring manufacturer, on the basis of these advantages and because of instances in which all rings produced from a poor routing were unacceptable, was led to consider a computer-aided routing system. The systematic approach outlined previously was followed in developing a routing program for automotive compression rings. Rings were classified according to cross-sectional shape, outer face design, and the need for a coating. Routing personnel were interviewed to determine what decisions were being made in producing a routing. Standard procedures and information tables were searched and quality control standards were reviewed. The sequence of operations was varied and the resulting changes in the product produced were noted. From this information, sequence lists were prepared specifying the preferred sequence of operations for each of some 24 defined classifications of compression rings. Based on ring classification and selected input information, a flowchart was prepared which determined the appropriate sequence of required operations. The required input data was selected through mathematical derivation of as many machine settings as possible and by determining what data was necessary in the calculations. Files were prepared containing casting and tooling inventories. The program was written in FORTRAN and was implemented on a 32K character Honeywell 1200 computer. Each of three reviews resulted in modifications, and the system was tested for a period of over a year on sample orders.

The example described here demonstrates the applicability of the proposed technique for developing a computer-aided routing.

ANALYSIS OF A SIMPLE SCHEDULING ALGORITHM FOR A MULTIPROCESSING COMPUTER SYSTEM

K. L. KRAUSE and V. Y. SHEN
Computer Sciences Department
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

This paper describes a simple algorithm to schedule a set of tasks with variable memory requirements but identical processing times. The underlying model is a multiprocessing computer system with n identical but independent processors and M units of main memory. Through deterministic analysis, the schedule produced by the algorithm is shown to be near-optimal in minimizing the time required to complete all the tasks.

I. INTRODUCTION

With the rapid expansion of computer structures to include large numbers of independent processors, we are increasingly concerned with the lack of good, basic algorithms to assign tasks to the processors in an efficient and acceptable manner. The purpose of this paper is to look at a simple model of computation, devise a scheduling algorithm for the model, and compare the length of such a schedule to an optimal schedule. The model considered here is a part of a more general model which is being investigated by the authors at Purdue University.

The basic model is n identical and independent processors available to complete k tasks. Each task requires a unit time slice and a known amount (pages) of memory. All tasks are available for scheduling initially. The goal of an optimal schedule (a schedule is defined as the description of the work done by each processor as a function of time) is to minimize the total time taken to complete all the given tasks.

The addition of the memory constraint to the model is a departure from similar models which have appeared in the literature (cf.[1,2,3]). The model deals with memory as an absolute constraint and not in a memory management sense (cf.[4]). We also assume that tasks may be transferred in and out of memory with negligible delay (cf.[5]).

The model can be considered as using a modified version of the round-robin scheduling discipline (cf.[6]) whereby all available tasks are given a single time slice in any one schedule period. The task list changes only at the completion of the schedule periods. Thus the optimal schedule that minimizes the schedule period has the equivalent effect of reducing the average response time.

An alternate view of the same basic model would be one of deadline scheduling where time is now the absolute constraint instead of memory. The requirement of each task would be a single processor for a given period of time. The goal would be to minimize the number of processors needed to guarantee the completion of all tasks by the given deadline.

The optimal schedule of the basic model of computation is known to exist due to the finiteness of the model. However, it is generally accepted that the generation of an optimal schedule requires an exhaustive process, which normally takes an exponential number of steps. This paper presents a scheduling algorithm which is computationally tractable (algebraic in the number of steps required) and produces a schedule that compares favorably to the optimal schedule through deterministic analysis.

II. ALGORITHM

The proposed algorithm assigns tasks from a task list ordered on memory requirement (m_j), largest requirement first. There are k tasks to be assigned. The tasks are assigned across time slices denoted by the index i. Each time slice has two associated variables, p_i - the number of processors in use and c_i - the amount of available memory. M is the total number of memory units at each time slice and n is the number of processors. N_a will denote the number of time slices used by the algorithm to assign the given tasks. The final value of N_a is determined iteratively attempting to place all the tasks in an incremental number of time slices.

```
0) Set $N_a = \lceil \max(k/n, \sum_{j=1}^{k} m_j/M \rceil$
```

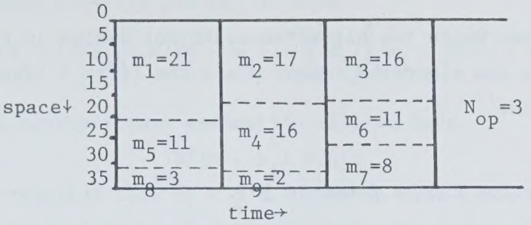
where $\lceil x \rceil$ represents the smallest integer greater than or equal to x.

```
1) For $i = 1,2,\ldots,N_a$ perform the following steps:
    1.1) Assign the ith task to the ith time slice.
    1.2) Set $p_i=1$ and $c_i=M-m_i$.
2) Set $j=N_a+1$
3) Set $I=\max(i \mid c_i \geq c_\ell$ for all $\ell$, $1 \leq \ell \leq N_a$ and $p_i \neq n)$
4) If $c_I < m_j$ then set $N_a=N_a+1$ and go to step 1.
5) Assign the jth task to the Ith time slice.
6) Set $p_I=p_I+1$, $c_I=c_I-m_j$, and $j=j+1$
7) If $j>k$ then terminate, otherwise go to step 3.
```

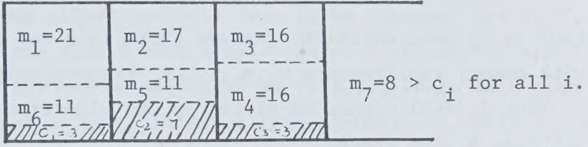
Variations of this simple algorithm can easily be devised which would remove the iterative aspect of total placement. This form of the algorithm lent itself well to the desired deterministic analysis and yet was algebraic in the number of steps required. At this point of development the tractability of analysis is more important than the ease of implementation on a computer system.

III. EXAMPLE

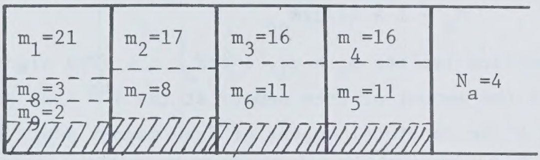
As an example let the task list ordered by memory requirements be J=(21,17,16,16,11,11,8,3,2), and the system parameters be k=9, n=3, and M=35. An optimal placement of tasks (schedule) would be denoted by:


0
5
10
15
20
25
30
35
space↓
$m_1=21$
$m_2=17$
$m_3=16$
$m_4=16$
$m_5=11$
$m_6=11$
$m_7=8$
$m_8=3$
$m_9=2$
$N_{op}=3$
time→

Here a vertical rectangular box represents a single time slice with 35 units of memory available and 3 processors. This could be considered as a two dimensional Gantt chart. The initial assignment by the algorithm uses 3 time slices which yields:


$m_1=21$
$m_2=17$
$m_3=16$
$m_6=11$
$m_5=11$
$m_4=16$
$c_1=3$
$c_2=7$
$c_3=3$
$m_7=8 > c_i$ for all i.

A second iteration of the algorithm using 4 time slices is required which yields:


$m_1=21$
$m_2=17$
$m_3=16$
$m_4=16$
$m_8=3$
$m_9=2$
$m_7=8$
$m_6=11$
$m_5=11$
$N_a=4$

In this small illustrative example we have a 33% difference between N_{op} and N_a, which is a worst case. Computer analysis of the algorithm using Monte Carlo and random task list generation indicates that in general the difference between N_{op} and N_a is around 5% of N_{op}.

IV. ANALYSIS

The focus in this paper is the deterministic analysis of the relationship which exists between the algorithmic solution, N_a, and the number of time slices required for the optimal schedule, N_{op}. Through deterministic analysis we can bound the relationship which exists between N_a and N_{op}. Notationally individual tasks will be referred to by noting their ordered position (index j) in the task list set J. J_1 is the largest task and J_k is the smallest. The size of J_j is m_j. The final schedule produced by the algorithm will be denoted by S and its length is N_a. The schedule produced by the previous, unsuccessful iteration will be denoted by S' and its length is $N_a - 1$. In order to establish the main theorem on the existing bound, the following lemma is used.

Lemma:

If $J_\ell (\ell \le k)$ is the task which the algorithm could not assign in the S' schedule and $J_\ell > M/4$, then the algorithm cannot place the first ℓ tasks in less than N_a time slices.

Proof:

A proof by contradiction can be found in [7].

Theorem:

Given the number of time slices required by the algorithm, N_a, and the number in the optimal solution, N_{op}, then $N_a < (4/3)N_{op} + 1$

Proof:

The proof is by contradiction. Assume there is a set of tasks J_1, $J_2, \ldots, J_k$ with memory requirements $m_1, m_2, \ldots, m_k$ respectively, in descending order. Task $J_\ell (\ell \le k)$ cannot be placed by the algorithm in schedule S' with $N_a - 1$ time slices where

$$N_a - 1 \ge (4/3)N_{op} \tag{1}$$

This assumption implies $m_\ell > c_i$, $1 < i < N_a - 1$. (The algorithm uses c_i to represent the amount of free memory at the i^{th} time slice.)

Let F and F' be the total amount of free memory space in the final schedule, S, and the next to last schedule, S', respectively. We have

$$F = N_a \cdot M - \sum_{j=1}^{k} m_j = \sum_{i=1}^{N_a} c_i \tag{2}$$

and

$$F' = (N_a - 1) \cdot M - \sum_{j=1}^{\ell-1} m_j < (N_a - 1) \cdot m_\ell, \tag{3}$$

as J_ℓ (with memory requirement m_ℓ) cannot be placed in any of the $N_a - 1$ time slices. It follows that

$$F = F' + M - \sum_{j=\ell}^{k} m_j < (N_a - 1)m_\ell + M - \sum_{j=\ell}^{k} m_j \tag{4}$$

From (2) and (4)

$$N_a \cdot M - \sum_{j=1}^{k} m_j < (N_a - 1)m_\ell + M - \sum_{j=\ell}^{k} m_j \tag{5}$$

$$N_a(M - m_\ell) < M - m_\ell + \sum_{j=1}^{\ell-1} m_j \tag{6}$$

$$N_a < 1 + \sum_{j=1}^{\ell-1} m_j/(M - m_\ell) \tag{7}$$

Since we also know as a lower bound that

$$N_{op} > \sum_{j=1}^{k} m_j/M \tag{8}$$

Therefore from (7) and (8) we have

$$(N_a - 1)/N_{op} < M/(M - m_\ell) \cdot \sum_{j=1}^{\ell-1} m_j / \sum_{j=1}^{k} m_j < M/(M - m_\ell) \tag{9}$$

By the contradictory assumption (1), we have

$$M/(M - m_\ell) > 4/3 \tag{10}$$

which requires that $m_\ell > \frac{1}{4} M$. Thus no time slice would contain more than three tasks in any schedule up to the placement of J_ℓ. The lemma established earlier allows us to consider the truncated set $J_1, J_2, \ldots, J_\ell$ for which N_a remains unchanged and N_{op} is non-increasing. The contradictory assumption (1) is still satisfied. Call the new optimal schedule length N'_{op}. The N'_{op} time slices can fall into three classes, say S, P, and T, where S is all time slices with single tasks, P all pairs, and T all triples. Visualize the optimal schedule as follows, where each time slice in class T is ordered so that the first task (denoted by class T_1) is at least as large as the other two assigned tasks.

S	P_1	T_1
	P_2	T_2
		T_3

In the above figure a subscripted class designator depicts a subset of that class with the appropriate number of tasks. The number of time slices in class S will be denoted by N_s, those in P by N_p, and those in T by N_t. Then we have

$$N'_{op} = N_s + N_p + N_t \tag{11}$$

and

$$\ell = N_s + 2N_p + 3N_t \tag{12}$$

In order to depict the worst case algorithmic schedule possible with these ℓ tasks, we will regroup the two P subclasses and the three T subclasses into classes H and G and note the algorithmic properties of these two new classes of tasks in forming the longest possible schedule.

$$H = \{\{P_1\}\}, \{P_2\}, \{T_1\}\}$$

$$G = \{\{T_2\}\}, \{T_3\} - J_\ell\}$$

Any time slice in the S' schedule (the next to final algorithmic schedule of length $N_a - 1$) beginning with a task from H must contain at least a pair, as J_ℓ, the smallest task in the truncated task set and the task which the algorithm cannot place in S', can coexist with any task of H. Any time slice in S' beginning with a task from G must contain at least a triple, as any two tasks no larger than the largest G task can coexist with J_ℓ.

This holds because the optimal schedule described has already demonstrated that each member of G can coexist with a task at least as large as itself and another task which is known to be at least as large as J_ℓ. The placing of a task of G by the algorithm dictates that there are no larger tasks yet to be placed.

The worst case algorithmic schedule can be broken into three cases depending on the number of tasks in class H and G denoted by N_H and N_G respectively, where

$$N_H = 2N_p + N_t, \tag{13}$$

and

$$N_G = 2N_t - 1, \tag{14}$$

$$N_s + N_H + N_G = \ell - 1 \tag{15}$$

Case I.

$$N_H = N_G \quad \text{or} \quad 2N_p = N_t - 1$$

The longest possible algorithmic schedule, S', is depicted as:

$$\begin{array}{c|c} S & H \\ & G \end{array}$$

where from (13)

$$N_a - 1 = N_s + N_H = N_s + 2N_p + N_t \tag{16}$$

from the contradictory assumption (1) and (11) we have

$$N_s + 2N_p + N_t \geq \frac{4}{3}(N_s + N_p + N_t) \tag{17}$$

it follows that

$$2N_p \geq N_s + N_t \tag{18}$$

which contradicts $2N_p = N_t - 1$.

Case II.

$$N_H > N_G \quad \text{or} \quad 2N_p > N_t - 1$$

The worst case S' would be:

$$\begin{array}{c|c|c} S & H_1 & H_2 \\ & G & H_3 \end{array}$$

then

$$N_a - 1 = N_s + N_H - (N_H - N_G)/2 \tag{19}$$

and it follows from (13), (14), and (19) that

$$N_a - 1 = N_s + N_p + (3/2)N_t - 1/2 \tag{20}$$

We have from (1) and (20)

$$N_s + N_p + (3/2)N_t \geq (4/3)(N_s + N_t + N_p) \tag{21}$$

it follows that

$$N_t \geq 2N_p + 2N_s + 3 \tag{22}$$

which contradicts

$$2N_p > N_t - 1.$$

Case III.

$$N_H < N_G$$

The longest S' schedule is:

$$\begin{array}{c|c|c} S & H & G_1 \\ & G_4 & G_2 \\ & & G_3 \end{array}$$

then

$$N_a - 1 = N_s + N_H + (N_G - N_H)/3 \tag{23}$$

or by (13) and (14)

$$N_a - 1 = N_s + 2N_p + N_t + (N_t - 2N_p - 1)/3 \tag{24}$$

which with (1) leads to

$$N_s + (4/3)(N_p + N_t) - 1/3 \geq (4/3)(N_s + N_p + N_t)$$

which is a contradiction for all N_s, N_p, and N_t.

Cases I, II, and III all contradict the assumption (1). Therefore it follows by way of contradiction that

$$N_a - 1 < (4/3)N_{op}.$$

We can demonstrate that this bound is approachable for a large data set by considering a task list with two classes of tasks, those with memory requirements approximately M/2 and those with requirements approximately M/4. By proper adjustment of the memory requirements we can construct an optimal schedule which is the same length as the lower bound and an algorithmic structure that approaches 4/3 of this length. The task sizes are chosen in such a manner that the optimal would use 100% of the memory resource and the algorithmic configuration would be the maximum possible, i.e. a pair in each time slice. Consider an arbitrary number of classes of time slices denoted by $A_1, A_2, \ldots, A_r$. The widths or number of time slices represented by each class will be denoted by $N_1, N_2, \ldots, N_r$ respectively. The optimal structure would then look as follows (all time slices within a denoted class are identical):

A_1	A_2	A_3	A_4		A_{r-2}	A_{r-1}	A_r
$M/2+\varepsilon$	$M/2-4\varepsilon$	$M/2-10\varepsilon$	$M/2-22\varepsilon$		$M/2-b\varepsilon$	$M/2-(2b+2)\varepsilon$	$M/4$
$M/2-\varepsilon$	$M/4+2\varepsilon$	$M/4+5\varepsilon$	$M/4+11\varepsilon$	··	$M/4+(b/2)\varepsilon$	$M/4+(b+1)\varepsilon$	$M/4$
	$M/4+2\varepsilon$	$M/4+5\varepsilon$	$M/4+11\varepsilon$		$M/4+(b/2)\varepsilon$	$M/4+(b+1)\varepsilon$	$M/4$
							$M/4$

where $\varepsilon > 0$

$$\text{and } b = 2^{r-2} + \sum_{q=1}^{r-4} 2^q \tag{25}$$

Note that the space requirement at each time slice is M. (i.e. all of memory.)

$$N_{op} = \sum_{i=1}^{r} N_i = \sum_{j=1}^{k} m_j/M \tag{26}$$

where m_j is the memory requirement for each of the k individual tasks.

We restrict ε at this point by

$$M/2 - (2b+2)\varepsilon > M/4 + (b+1)\varepsilon \tag{27}$$

to guarantee the first tasks in A_{r-1} are larger than the others

$$\text{or} \qquad \varepsilon < M/12(b+1) \; . \tag{28}$$

This restriction is imposed so we can explicitly depict the algorithmic structure. The final algorithmic structure, S, would be:

$$\begin{array}{l|l|l|l|c|l|l}
M/2+\varepsilon & M/2-\varepsilon & M/2-4\varepsilon & M/2-10\varepsilon & & M/2-b\varepsilon & M/2-(2b+2)\varepsilon \\
M/4 & M/4+2\varepsilon & M/4+5\varepsilon & M/4+11\varepsilon & .. & M/4+(b+1)\varepsilon & M/2-(2b+2)\varepsilon
\end{array}$$

By looking at the number of tasks associated with each subclass above we have:

$$N_a = N_1+N_1+N_2+N_3+\ldots+N_{r-2}+N_{r-1}/2 \tag{29}$$

In order to satisfy this exact schedule we must have the following relationships.

$$\begin{aligned}
N_r &= N_1/4 \\
N_2 &= N_1/2 \\
N_3 &= N_2/2 = N_1/2^2 \\
N_4 &= N_3/2 = N_1/2^3 \\
&\vdots \\
N_{r-1} &= N_{r-2}/2 = N_1/2^{r-2}
\end{aligned}$$

It then follows from (26) that

$$N_{op} = \sum_{i=1}^{r} N_i = N_1\left(\sum_{i=0}^{r-2} \left(\tfrac{1}{2}\right)^i + \tfrac{1}{4}\right) \tag{30}$$

and from (29) that

$$N_a = N_1\left(2 + \sum_{i=1}^{r-3} \left(\tfrac{1}{2}\right)^i + \left(\tfrac{1}{2}\right)^{r-3}\right) \tag{31}$$

let $r \to \infty$, it follows from (30) and (31) that

$$N_{op} \to (9/4)N_1$$

d $\quad N_a \to 3N_1$

therefore $\quad N_a \to (4/3)N_{op}$

The only remaining case to consider for the completion of the theorem proof is the case where in the S' schedule there are a large number of time slices with $p_i = n$ (all the processors being used) and $c_i > m_\ell$ (the processor restriction was all that prevented the placement of J_ℓ). Since the nature of the algorithm is to achieve uniform usage of processors and memory at each time slice, this situation did not occur in our simulation analyses by computer and was not expected. It is possible to construct a formal proof by exhaustive elimination of cases for a given processor restriction. The authors have formalized proofs for restrictions up to six processors using this method.

V. CONCLUSION

We have demonstrated the feasibility of deterministic analysis for a simple yet non-trivial model of scheduling. Although the model may have only a few direct applications at this time, it is expected that the technique of using both computer simulation and deterministic analysis can be applied to other variations of this model. This approach is being pursued by the authors at Purdue University.

REFERENCES

1. Graham, R. L. "Bounds on Multiprocessing Timing Anomalies," SIAM J. on Applied Math. (17,2) Mar. 1969.

2. Kleinrock, L. "A Continuum of Time-Sharing Scheduling Algorithms," SJCC (36) 1970.

3. Muntz, R. R., Coffman, E. G., Jr. "Preemptive Scheduling of Real Time Tasks on Multiprocessor Systems," JACM (17,2) Apr. 1970.

4. Denning, P. J. "The Working Set Model for Program Behavior," CACM (11,5) May, 1968.

5. Chen, Y. E., Epley, D. L. "Memory Requirements in a Multiprocessing Environment," JACM (19,1) Jan. 1972.

6. Coffman, E. G., Jr., Kleinrock, L. "Computer Scheduling Measures and their Countermeasures," SJCC (32) 1968.

7. Krause, K. L., Shen, V. Y. "Analysis of a Simple Scheduling Algorithm for a Multiprocessing System," Technical Report No. 74, Computer Sciences Department, Purdue University, September 1972.

SCHEDULING IN A COMPUTER SYSTEM WITH LIMITED MEMORY

L. S. CHEUNG
School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

Y. E. CHEN*
Bell Telephone Laboratories
Columbus, Ohio 43213

ABSTRACT

The problem of scheduling jobs in a batch processing computer system with limited main memory size and finite channel rate is considered. The model used consists of a limited main memory and a large auxiliary memory, linked by a channel. At first, all the jobs are assumed to reside in the auxiliary memory and have to be loaded into the main memory before they can be executed. Due to the limitations on the main memory size and the channel rate, the completion time for a given set of jobs varies drastically as the sequence of loading changes. This effect is studied in this paper and loading strategies are developed that would reduce the completion time under different situations.

I. INTRODUCTION

This paper investigates the problem of scheduling in a batch processing computer system. The model considered has the form shown in Fig. 1. The system has a large auxiliary memory that holds all the jobs to be processed. The jobs have to be moved one by one into the main memory to be processed by the processor. The main memory has a limited size M. A channel is used as the link between the two memories.

Let the resource request of each job in the system be represented by a two-tuple (m, t), where m is the main memory requirement and t is the total processing time. If the channel has a finite rate C for data transfer, it takes m/C time units to load a job for execution. The problem is to determine a sequence for loading jobs from the auxiliary memory so that the total time required to process these jobs is minimized. It is assumed that there is negligible interference between the processor and the channel. This can be achieved when interleaved memory is used. A similar model was investigated recently by Chen and Epley [1]. They have determined the feasible memory size and channel rate for a known job schedule. This paper describes the results on the determination of two optimal schedules under two different situations when the memory size and the channel rate are known. These two algorithms can be extended to the general case with good result. This is shown using simulation.

The following examples illustrates the variation on completion time with different execution sequences. Given a system with M = 10, C = 1 and two jobs with resource requests $T_1 = (7,2)$ and $T_2 = (5,10)$, the sequence (T_1,T_2) has a termination time of 22 (Fig. 2a). If T_2 were executed first, although it takes a span of 12 time units to load T_1 due to insufficiency of memory space, the termination time is 19 (Fig. 2b). For the same system and four jobs with resource requests $T_1 = (1,9)$, $T_2 = (1,9)$, $T_3 = (9,1)$ and $T_4 = (9,1)$, the sequence (T_1,T_3,T_2,T_4) provides the minimum termination time 21 (Fig. 3a). A different loading sequence (T_1,T_2,T_3,T_4) has termination time 29 (Fig. 3b). When M = 18 or more, the termination time should be 21 (Fig. 3c).

---

*On leave from the staff of the School of Electrical Engineering, Purdue University.

A previous result by Johnson [2] shows that the optimal sequencing of a set of jobs which demand for a certain amount of services of A processor followed by a certain amount of services of B processor can be determined by a simple comparison procedure. Let a job be denoted by a two-tuple (a, b), then job i should definitely precede job j for early termination if

$$\text{MIN}\ (a_i,\ b_j) \le \text{MIN}\ (a_j,\ b_i). \tag{1}$$

This result fits our model if the memory has "sufficient" size. To be exact, the memory will not cause delay during processing. In Fig. 3c, the optimal sequence as determined by the slightly modified Eq. 1,

$$\text{MIN}\ (m_i/c,\ t_j) \le \text{MIN}\ (m_j/c,\ t_i), \tag{2}$$

is $(T_1,\ T_2,\ T_3,\ T_4)$. With memory size greater than or equal to 18, the loading sequence provides the minimum total execution time.

In this paper, we model the general problem into a set of non-linear recursive equations. There is no analytical solution available. However, by adding some constraints, the equations can be solved. Through simulation, we show that the optimal sequencing obtained from the restricted solutions is a very nice one when it is applied to a less restricted environment.

II. THE MODEL

A sequence L is formed to execute a set of n jobs. Let d(j) be the information that has to be loaded such that the jth job can be processed, and K(j) be the useful information stored in memory right after the completion of the jth job in sequence L. Based on the criterion that a job cannot be started unless the whole job is currently in the memory, the following equations should be true.

$$d(j) = \text{MAX}[0,\ m(j) - K(j-1)] \quad \text{for } j = 1 \text{ to } n \tag{3}$$

$$K(j) = K(j-1) + d(j) - m(j) + \text{MIN}[Ct(j),\ M - K(j-1) - d(j)] \quad \text{for } j = 1 \text{ to } n \tag{4}$$

$$K(0) = 0 \tag{5}$$

$$W = \sum_{j=1}^{n} t(j) + \sum_{j=1}^{n} [d(j)/C] \tag{6}$$

W is the total duration of the execution. Eq. 3 shows that the amount of information should be added is the difference between the information required and the information currently stored in the memory. Eq. 4 shows the amount of information at the end of the jth execution is the remnant of the (j-1)th execution plus the amount transmitted into the memory and deducted by m(j) through terminating of the jth job. Eq. 5 is the initial condition such that there is no preloading at the time when execution starts. Eq. 6 shows the total execution time which is the sum of all the job execution times plus the channel transmit times required to enable the subsequent executions. $\sum_{j=1}^{n} t(j)$ is the total execution duration of the set of jobs, therefore it is a fixed quantity. In order to minimize W, we should find a loading (or execution) sequence such that $\sum_{j=1}^{n} d(j)$ is minimum. Let this sequence be denoted as L_0. Due to the non-linearity of the equations, a general solution cannot be obtained. Here, we have obtained some particular solutions with added restriction imposed on the equations. From Eq. 4,

$$K(j) = \begin{cases} M - m(j) & \text{if } Ct(j) \geq M-K(j-1) - d(j) \quad (7) \\ & \text{for } j = 1 \text{ to } n \\ K(j-1) + d(j) - M(j) + Ct(j) & \text{if } Ct(j) < M-K(j-1) - d(j) \quad (8) \\ & \text{for } j = 1 \text{ to } n \end{cases}$$

If the condition in Eq. 7 is met, it is equivalent to that the memory capacity is the major handicap which delays the execution. On the other hand, if the condition in Eq. 8 is met, it is equivalent to that the channel is the major draw back which delays the execution. The later case is exactly the one illustrated by Johnson [2], and its optimal sequence can be defined by Eq. 2. Conversely, to justify whether a sequence generated by Eq. 2 is optimal, we can simply test whether the IF condition in Eq. 8 is satisfied for all the j's. Or, we can verify by employing the graphical procedure using the generated sequence as shown in Fig. 1 and 2.

Now, consider the condition as shown in Eq. 7. We can substitute $K(j) = M-m(j)$ into Eq. 3 and then Eq. 6, then the following equations are obtained.

$$W = \sum_{j=1}^{n} t(j) + (1/c) \{m(1) + \sum_{j=2}^{n} \mathrm{MAX}[0,m(j) + m(j-1) - m]\} \qquad (9)$$

or

$$W = \sum_{j=1}^{n} t(j) + (1/c) \sum_{j=1}^{n} \mathrm{MAX}[0,m(j) + m(j-1) - M] \qquad (10)$$

where

$$m(0) = M \qquad (11)$$

The following theorem shows the optimal sequence which minimizes Eq. 10.

Theorem The optimal sequence L_0 which minimizes W with added constraint as shown in Eq. 7 has the following relation.

$$m(1) \leq m(n-1) \leq m(3) \leq m(n-3) \leq \ldots. m(4) \leq m(n-2)$$
$$\leq m(2) \leq m(n) \qquad (12)$$

Proof. See Appendix.

To test whether the condition in Eq. 7 is satisfied, we can construct the sequence according to the theorem, then test whether the IF condition in Eq. 7 is satisfied for all the j's. Or, we can employ the graphical procedure as shown in Fig. 1 and 2. A simplified test is to check whether

$$Ct(j) \geq M - m(j) \qquad (13)$$

because

$$\begin{aligned} Ct(j) &\geq M - K(j-1) - d(j) \\ &= M - K(j-1) - MAX[0, m(j) - K(j-1)] \\ &= MIN\ [M - K(j-1),\ M - m(j)]. \end{aligned}$$

Therefore, if Eq. 13 is satisfied for all the jobs, the sequence as defined in the theorem should provide the minimum overall execution interval.

We have obtained two optimal sequences for two restricted cases. The question how good they will perform when the IF conditions in Eq. 7 or 8 are not satisfied. In other words, the set of jobs is so diversified such that the sequence generated by Eq. 2 does not satisfy the IF condition for all j's as shown in Eq. 8, and the sequence generated by the theorem does not satisfy the IF condition for all j's as shown in Eq. 7. The following simulation shows that the sequence generated by the theorem is a better choice.

III. THE SIMULATED RESULT

Let the sequence generated by Johnson's formula (Eq. 2) be denoted as sequence B; the sequence generated by the theorem as sequence A. Also, we consider a third procedure which makes use of an ordering relation considering only two jobs; and name the resulted sequence as sequence C. For a set of jobs, let $\bar{m}$ be denoted as the mean memory requirement of a set of jobs, and t be the mean processing time of a set of jobs. The two are expressed in terms of M and (M/C).

For each combination of $\bar{m}$ and $\bar{t}$, a total of 100 sets of jobs are generated randomly, and each set has at least 10 jobs and may have as many as 50. For each set of jobs, 25 or 26 randomly generated sequences are formed. In Table 1, we compare them with sequence A, B, and C. We define

$$T^i = MIN\ \{T^i_A, T^i_B, T^i_C\} \qquad (14)$$

where T^i_X is the completion time associated with the set i and sequence X (A, B, or C).

In Table 1, all the completion time for the randomly generated sequence are normalized by T^i. A randomly generated sequence from set i is called a failure if its completion time is shorter than T^i. For a given set of jobs i, a win for a sequence X is when $T^i = T^i_X$. Since two sequences may generate the same completion time, therefore, it is possible for two sequences both score a win.

For example, when $\bar{m} = (3/8)\ M$ and $\bar{t} = 2M/3C$ with 100 sets of jobs, we generate 25 random sequences plus sequence A, B and C for each set of jobs. There are 101 failures which is 4.04% of the 2500 sequences. The average normalized completion time for the failures is .98691, and the average normalized completion for all the sequences is 1.04692. Sequence A scores 99 wins, and sequence B scores two wins.

From the simulation, we conclude that sequence A is a good sequence to be followed.

IV. CONCLUSION

In this paper, we have generated a model to describe the effect of loading sequence toward the overall completion time. Due to the nonlinearity of the recursive equations, we cannot obtain a general solution. However, two restricted solutions have been obtained. The two sequences are tested through simulation in a rather general environment. The one developed in this paper provides good result. This leads the authors to conclude that the sequence should be seriously considered during application.

APPENDIX

In this part, a formal proof of the theorem is presented. $\sum_{j=1}^{n} t(j)$ is a fixed quantity, therefore the sequence which minimizes $\sum_{j=1}^{n} \mathrm{MAX}[0, m(j) + m(j-1) - M]$ minimizes W. Let the memory requirements be subscribed according to their magnitude such that m_1 is the smallest memory requirement among the set of jobs, and M_n is the largest memory requirement of the set of jobs. If the theorem is true, the ordering of jobs according to their memory requirements is

$$M\ m_1\ m_{n-1}\ m_3\ m_{n-3} \cdots m_4 m_{n-2}\ m_2\ m_n. \qquad \text{(A)}$$

Since m(1) (the first element to be loaded and executed) can only be loaded at time zero, it is equivalent to assume that the job executed prior to the first job has memory requirement M.

Let Q(i,j) be an operation to invert the subsequence as defined by the starting entry m_i and the terminating entry m_j, i.e.

$$\ldots m_a m_i m_b \ldots\ldots m_c m_j m_d \ldots\ldots \xrightarrow{Q(i,j)} \ldots m_a m_j m_c \ldots m_b m_i m_d \ldots\ .$$

Here, we want to show that sequence A is the sequence which minimizes W.

Lemma 1 For a given set of m, one of the general characteristic of the optimal sequence is that m_n should always be loaded last.

Proof. Assume there is an optimal sequence

$$M\ m_a \ldots\ldots m_b m_n m_c \ldots m_d.$$

The operation Q(n,d) will alter the original sequence into the following sequence

$$M\ m_a \ldots m_b m_d \ldots m_c\ m_n.$$

The new sequence will provide a smaller W because $m_n > m_d$ and $\mathrm{MAX}[0, m_b + m_n - M] \geq \mathrm{MAX}[0, m_b + m_d - M]$. Contradiction occurs, so the Lemma should be true.

Lemma 2 In the optimal sequence, any two pairs of adjacent elements, say $[m_a, m_b]$ and $[m_c, m_d]$, should either be

$m_a > m_d$ and $m_b < m_c$

or

$m_a < m_d$ and $m_b > m_c$,

Proof. Assume, on the contrary, that $m_a > m_c$ and $m_b > m_d$. The operation Q(b,c) will generate a sequence with smaller W, because

$$\mathrm{MAX}[0, m_a + m_b - M] + \mathrm{MAX}[0, m_c + m_d - M] \geq \mathrm{MAX}[0, m_a + m_c - M] + \mathrm{MAX}[0, m_b + m_d]$$

and the rest of the sequence remains unchanged. Again, contradiction validate the Lemma.

In Lemma 1 and Lemma 2, we have shown that an optimal sequence should possess the properties as stated in Lemma 1 and 2. The following lemma shows that sequence A which possesses the two properties is an optimal sequence.

Lemma 3 Sequence A is an optimal sequence.

Proof. Assume there is an optimal sequence, say B, which is different to Sequence A; however m_n should be the last in both sequences according to Lemma 1. Further, assume the two sequences have same subsequences from M to m_a and m_b to m_n. We want to show that the next element to the right of m_a and the element to left of m_b in B are the same element as shown in A. If this is true, mathematical induction tells us that sequence A is truly sequence B itself.

Sequence B is shown as follows,

$$M \ldots m_a m_c \ldots m_d m_e m_f \ldots m_g m_b \ldots m_n \qquad \text{(B)}$$

The m's are the subscribed variables. Since the subsequence of B from M to m_a is the same as A, m_a should either be larger than or smaller than all the elements between m_a and m_b. Assume m_a is the larger quantity, and m_a is greater than m_b. m_e is the smallest among the set. Since $m_a > m_f > m_e$ and $m_a > m_c > m_e$, Q(c,e) will not increase W according to Lemma 2. If $m_b > m_a$, Q(e,g) after Q(c,e) will further shorten W. By the same token, if m_a is the smaller quantity than any of the elements between m_a and m_b, W will be reduced when the Q operation moves the largest element between m_a and m_b to take the place next to m_a. Further reduction can be realized to move the largest element to the left of m_b if $m_a > m_b$.

The above shows that the elements adjacent to m_a and m_b of sequence B are truly the same elements as shown in sequence A. By mathematical induction, sequence A is in reality the optimal sequence B.

REFERENCES

1. Chen, Y. E. and D. L. Epley, "Memory Requirement in a Multiprocessing Environment," JACM 19, 1, Jan. 1972, pp. 57-69.

2. Johnson, S. M., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Nav. Res. Log. Quart. 1, No. 1, March 1954.

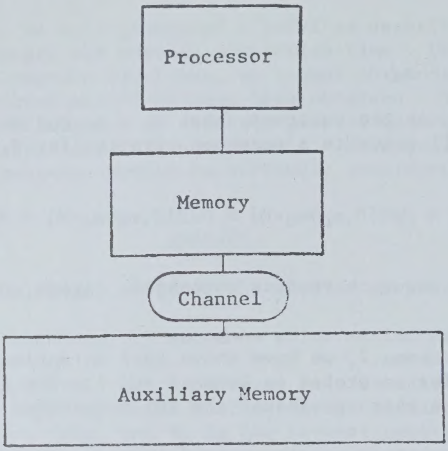

Processor
Memory
Channel
Auxiliary Memory

Fig. 1

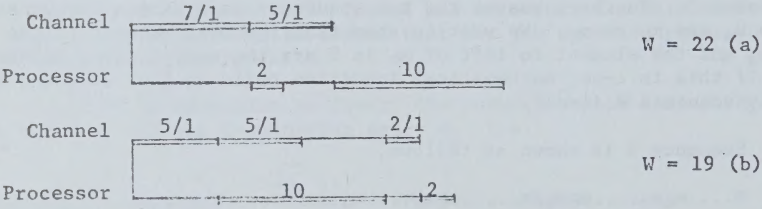

Channel
7/1
5/1
Processor
2
10
W = 22 (a)
Channel
5/1
5/1
2/1
Processor
10
2
W = 19 (b)

Fig. 2

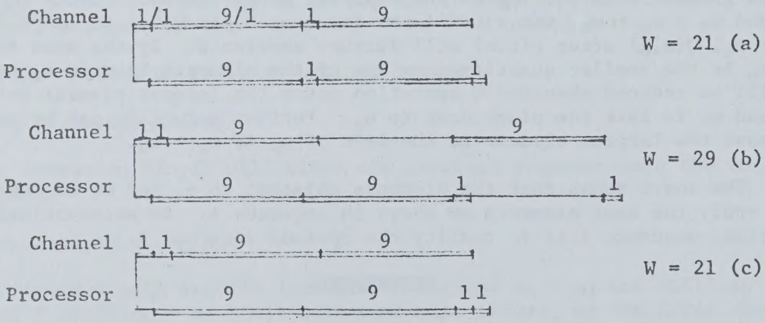

Channel
1/1
9/1
1
9
Processor
9
1
9
1
W = 21 (a)
Channel
1 1
9
9
Processor
9
9
1
1
W = 29 (b)
Channel
1 1
9
9
Processor
9
9
1 1
W = 21 (c)

Fig. 3

TABLE 1

$\overline{M}$ (M)	$\overline{t}$ (M/C)	No. of sets of jobs	Total no. of sequences generated	Total no. of failures	% of failures	Average normalized completion time for the failures	Average normalized completion time	No. of wins A	No. of wins B	No. of wins C
1/2	2/3	100	2500	10	0.4%	.986554	1.08707	100	-	-
1/2	1/2	100	2500	22	0.88%	.991023	1.08641	100	-	-
1/2	3/8	100	2500	49	1.96%	.986927	1.06549	100	-	-
1/2	1/4	100	2500	44	1.76%	.996345	1.04345	96	7	3
3/8	3/4	100	2600	115	4.423%	.990807	1.04327	99	1	1
3/8	2/3	100	2500	106	4.24%	.989871	1.04673	99	2	2
3/8	2/3	100	2600	118	4.53846%	.989647	1.04726	100	-	-
3/8	2/3	100	2500	101	4.04%	.986910	1.04692	99	2	-
3/8	1/2	100	2600	151	5.80769%	.984839	1.05612	100	-	-
3/8	1/2	100	2500	131	5.24%	.985038	1.06023	100	1	-
3/8	3/8	100	2600	133	5.11538%	.980932	1.06647	96	4	2
3/8	3/8	100	2500	198	7.92%	.979679	1.06078	97	2	-
1/4	1	100	2500	81	3.24%	.995931	1.01022	82	69	67
1/4	3/8	100	2500	63	2.52%	.995679	1.01213	81	63	63
1/4	2/3	100	2500	150	6.0%	.993278	1.01599	88	54	52
1/4	1/2	100	2500	213	8.52%	.991729	1.02194	81	39	39
1/4	3/8	100	2500	267	10.68%	.990273	1.02957	83	30	27
1/4	1/4	100	2500	441	17.64%	.984100	1.03632	84	19	17

DISJOINT REPRESENTATION OF TREE REALIZABLE SEQUENCES

SUKHAMAY KUNDU
IBM Thomas J. Watson Research Center , Yorktown Heights, New York

ABSTRACT

A necessary and sufficient condition is obtained for the edge disjoint tree realization of two tree realizable degree sequences. The condition is simply that the sum sequence be graphical.

INTRODUCTION

We shall consider undirected graphs without multiple edges and loops. A realizing graph of a (degree) sequence $<d_i : 1 \le i \le N>$ is a graph whose ith vertex v_i has degree d_i. A necessary and sufficient condition for edge disjoint realization of two degree sequences was obtained in [5]. We had assumed that either their sum sequence or one of the given degree sequences is almost regular, i.e., the maximum and the minimum degrees differ by no more than one. Later Kleitman and Wang [4] have shown that the condition of almost regularity can be "relaxed" at a single vertex. In the present note we prove that two arbitrary tree realizable degree sequences can in fact be realized as edge disjoint trees provided that their sum sequence is graphical. An interesting aspect of this theorem is that each of the two "parts" in the edge disjoint realization is connected. In the earlier works [4] ,[5] the question of connectedness of the "parts" was ignored and remains to be investigated (See [8]).The realization of graphs of given degree sequences having many edge disjoint (spanning) trees is important for several reasons. For example, a graph having two such trees will be favorable to the short player in Shannon's switching game, independent of the pair of nodes chosen in the game. In addition, if there are n edge disjoint trees then edge connevtivity of the graph is at least n (though the converse is not true). From the point of view of edge connectivity alone Edmond's Theorem is the best possible [1].

MAIN RESULTS

We prove the following

Theorem 1.1 Let $<d_i^{(j)}: 1 \le i \le N>$, j=1,2 be two sequences of positive integers where for each value of j the sum of degrees $d_i^{(j)}$ equals 2(N-1). Then there exists edge disjoint tree realization of the two sequences if and only if the sum sequence $<d_i^{(1)} + d_i^{(2)}>$ is graphical.

The proof of (1.1) depends upon few preliminary lemmas. Recall that a sequence of positive integers $<d_i>$ is tree realizable if and only if $\Sigma d_i = 2(N-1)$. Theorem 1.1 is trivially true for N=4. In the following we shall assume $n \geq 5$. The vertex set $V = \{v_1, v_2, .., v_n\}$ is fixed throught the discussion.

LEMMA A Let $d_1 \geq d_2 \geq \ldots \geq d_N = 1$ be a graphical sequence of N terms where $N - 2 \geq d_1 \geq d_{N-1} \geq 2$ and $\Sigma d_i = 4N - 6$. Let $d_j \geq 3$ and $d'_i = d_i - \delta(i,j)^*$ for $1 \leq i \leq N-1$. The sequence $<d'_i>$ is graphical unless

$$<d_i> = <d_1, d_2, d_3, 3, 2, \ldots, 2, 1> , \ j=4 \tag{A.1}$$

Proof. Suppose the sequence $<d'_i : 1 \leq N-1>$ is not graphical. It follows from Erdős-Gallai Theorem that one of the inequalities

$$\sum_{i \leq p} d'_i \leq p(p-1) + \sum_{i > P} \min (p, d'_i) \tag{A_p}$$

$1 \leq p \leq N-1$ is false.** That (A_p) is true for $p = 1$ and $p \geq j$ follows from the hypothesis. The falsity of (A_p), $1 < p < j$ implies

$$\begin{aligned} 4N - 6 &= \sum_{i \leq p} d'_i + \sum_{i > p} d'_i + 2 \\ &> p(p-1) + 2 + 2 . \sum_{i > p} \min (p, d'_i) \\ &\geq p^2 - p + 2 + 4 (N-p-1) \end{aligned}$$

or

$$p^2 - 5p + 4 < 0$$

Thus $p = 2$ or 3. But $d_1 + d_2$ being at most $2(N-2)$, $p = 3$ is the only possibility and in that case

$$S = d_1 + d_2 + d_3 > 3.2 + 2 (N-4) = 2N - 2.$$

The sequence $<d_i>$ is therefore either as in (A.1) or one of the following:

$$<d_1, d_2, d_3, 4, 2, 2, \ldots, 2, 1>; \ S = 2N - 1, \ j=4.$$

$$<d_1, d_2, d_3, 3, 3, 2, \ldots, 2, 1>; \ S = 2N - 1, \ j=5$$

But the sequences $<d'_i>$ corresponding to the last two sequences satisfy (A_3). The lemma is proved.

We shall need another lemma similar to the above.

---

* $\delta(i,j)=0$ for $i \neq j$, $\delta(j,j)=1$.

** without loss of generality, $<d'_i>$ can be assumed to be non-increasing; also in lemma B.

LEMMA B Let $d_1 \geq d_2 \geq \ldots \geq d_N$ be a graphical sequence such that $\sum d_i = 4(N-1)$ and $d_N \geq 3$. Then for all $j \leq N-1$, the sequence $\langle d_i' = d_i - \delta(i,j) : 1 \leq i \leq N-1 \rangle$ is graphical provided that $d_1' \leq N-2$.

Proof. Note that for $d_1 = N-1$ we have $d_i = 3$, $i \geq 2$ and $j=1$, and the lemma is true. Assume now $d_1 < N-1$. If the inequality (A_p) is violated then adding $\sum_{i>p} d_i'$ to both sides, we get

$$4(N-1) - 4 > p(p-1) + \sum_{i>p} \min(p, d_i') + \sum_{i>p} d_i \tag{B.1}$$

For p=2, we have from (B.1) $N < 6$ and further $\langle d_i \rangle = \langle 4, 3, 3, 3, 3 \rangle$ in which case the lemma is true. For $p \geq 3$, (B.1) can be simplified to

$$p^2 - 7p + 2N < 0$$

which is absurd since N can be assumed > 6.

We are ready to prove our Theorem.

Proof of Theorem. (By induction on N.) Let $d_i = d_i^{(1)} + d_i^{(2)}$. Without loss of generality $\langle d_i \rangle$ can be assumed to be non-increasing. Clearly $2 \leq d_N \leq 3$ and $d_1 \geq 4$. We shall write T_j, j=1,2 for a tree realizing the degree sequence $\langle d_i^{(j)} \rangle$.

Case 1. $d_N = 2$.

Let $d_1^{(1)} \geq 2$. If $d_2 = N-1$ then one of the following is true: (a) $d_2^{(2)} \geq 2$, (b) $d_2^{(2)} = 1$, $d_2^{(1)} = N-2$, $d_1^{(2)} \geq 2$. Otherwise, $d_1^{(1)} + d_2^{(1)} = 2(N-2)$ which is not true. In each case we can choose such j and k that $\{j,k\} = \{1,2\}$ and $d_1^{(j)} \geq 2$, $d_2^{(k)} \geq 2$. Apply induction hypothesis to the sequences

$$\langle d_1^{(j)} - 1, d_2^{(j)}, \ldots, d_{N-1}^{(j)} \rangle \tag{1.1.1}$$

$$\langle d_1^{(k)}, d_2^{(k)} - 1, \ldots, d_{N-1}^{(k)} \rangle.$$

whose sum sequence $\langle d_1 - 1, d_2 - 1, d_3, \ldots, d_{N-1} \rangle$ is graphical. Obtain edge disjoint realizing trees T_j', T_k' respectively, on the vertices $V - v_n$. Write $T_j = T_j' + (v_1, v_N)$, $T_k = T_k' + (v_2, v_N)$ for the required trees. Let us now assume that $d_2 \leq N-2$. Choose $k \geq 2$ such that $d_k^{(2)} \geq 2$. If the sequence $\langle d_1 - 1, d_2, \ldots, d_{k-1}, d_k - 1, d_{k+1}, \ldots, d_{N-1} \rangle$ is graphical, then we can use the construction as before with j=1 and v_k playing the role of v_2. On the other hand if no such k exists, then in view of lemma A, $d_i = 2$ for all $i \geq 5$, $d_4 = 3$ and k being equal to 4

$d_2^{(2)} = d_3^{(2)} = 1$. It follows that each of $d_1^{(2)}, d_2^{(1)}$ is greater than one, and we are back to (1.1.1).

Case 2. $d_N = 3$.

We assert that for some i, j, k, $j \neq k$ the following are true:

(a) $d_j = 3$

(b) $d_j^{(i)} = 1,\ d_k^{(i)} > 1$

(c) The sequence obtained by removing jth term in $<d_t - \delta(k,t) : 1 \leq t \leq N>$ is graphical.

Once we prove the assertion the realizing trees are obtained as follows. Suppose, for simplicity of notation, i=1, and j=N. By the induction hypothesis let T_1', T_2' be a pair of edge disjoint realizing trees for sequences

$$<d_1^{(1)}, d_2^{(1)}, \ldots, d_{k-1}^{(1)}, d_k^{(1)} - 1, d_{k+1}^{(1)}, \ldots, d_{N-1}^{(1)}>$$

$$<d_1^{(2)}, d_2^{(2)}, \ldots, d_{N-1}^{(2)}>$$

on the vertices $V - v_N$ in that order. Let (v_p, v_q) be an edge of T_2' which is not adjacent to v_k. Define $T_1 = T_1' + (v_N, v_k)$ and $T_2 = T_2' + (v_N, v_p) + (v_N, v_q) - (v_p, v_q)$. To prove the assertion. If $d_1 \leq N-2$, take j=N and i,k arbitrarily according to (b). By lemma B, the condition c) is satisfied. For $d_1 = N-1$, suppose $1 < d_1^{(1)} = N-p$ and for every $d_j^{(1)} = 1$, $d_j \geq 4$ (otherwise k=1, i=1 and j are "good"). Since there are at least (N-p) of $d_i^{(1)}$ each equal to one,

$$2(N-1) = \sum d_i^{(2)} \geq 3(N-p) + p$$

or

$$p \geq \frac{N+2}{2}.$$

In other words, $d_1^{(2)} \geq 2$. Choose i=2, k=1 and j=N. The proof of theorem 1.1 is complete.

Theorem 1.1 cannot be generalized to 3 or more sequences even under the hypothesis that the sum sequence of subfamilies of the sequences are graphical. See example 1.3. A useful corollary to our theorem is stated as follows.

Corollary 1.2 Let $d_1 \geq d_2 \geq \ldots \geq d_N$ be a graphical sequence such that the sum of the d_i is at least 4(N-1) and $d_N \geq 2$. Then there is a realization resulting in a graph containing two disjoint spanning trees.

Corollary 1.2 is much "stronger" than the best result known so far. Edmonds [1] has shown that for $d_N \geq p$ there exists a p-edge connected realization of $<d_i>$. In the present case p=2. Existence of two edge disjoint trees certainly imply that the edge connectivity ≥ 2. However, the converse is not true.

Proof of Corollary (by induction). The corollary is clearly true for $N=4$. Suppose it is true for sequences with $N-1$ or less terms. Let $d_N = k$. Consider the sequence $D' = \langle d'_i : 1 \leq i \leq N-1 \rangle$ where $d'_i = d_i - 1$ if $i \leq k$ and d_i for $k < i \leq N-1$. D' is graphical and the sum of the degree is $\geq 4(N-2)$ except possibly when $k=2,3$. However, for $k=2$, $\sum d'_i = \quad d_i - 4 \geq 4(N-2)$ and for $k=3$, $\sum d'_i = \sum d_i - 6 \geq 4(N-2)$ unless $\sum d_i = 4(N-1)$ in which case theorem 1.1 applies with $d_i^{(1)} = \left[\frac{d_i}{2}\right]$, adding 1 to it for half of the places where d_i is odd. Thus D' satisfies the corollary and a realizing graph G' can be obtained containing two edge disjoint spanning trees T'_1, T'_2. Define $G = G' \cup \{(v_i,v_N) : 1 \leq i \leq k\}$ and $T_1 = T'_1 \cup (v_1,v_N)$, $T_2 = T'_2 \cup (v_2,v_N)$.

Example 1.3. Consider the tree realizable sequences as in Figure 1. It is not possible to realize all of the sequences by trees no two of which have an edge in common. Because the edge incident with vertex v_7 that belongs to the graph G_3 realizing the third sequence is necessarily adjacent to one of the vertices v_1,v_2 and v_3, and thus G_3 is not a tree. The sum sequence of any two of the three sequences is graphical and thus has edge disjoint representation

$\langle 2, 3, 2, 2, 1, 1, 1 \rangle$ Sum sequence
$\langle 3, 2, 3, 1, 1, 1, 1 \rangle$ $= \langle 6, 6, 6, 5, 5, 5, 3 \rangle$
$\langle 1, 1, 1, 2, 3, 3, 1 \rangle$

Fig. 1

by theorem 1.1. For each $N \geq 8$ it is easy to construct from Fig. 1 three tree realizable sequences of length N whose sum sequence is graphical but has no edge disjoint tree realization. An edge disjoint realization of the sequences in Fig. 1 exists in which the first two graphs are trees.

Nevertheless, a parallel of Theorem 1.1. holds true for three sequences provided the minimum of $d_i^{(1)} + d_i^{(2)} + d_i^{(3)}$ is at least 4. This will be discussed in a forthcoming paper.

REFERENCES

[1] Edmonds, J., "Existence of k-edge connected ordinary graphs with prescribed degrees," J. Res. Nat. Bur. Stand., Sect. B68 (1964), 73-74.

[2] Edmonds, J., "Lehman's Switching Game and a Theorem of Tutte and Nash-Williams," J. Res. Nat. Bur. Stand., 69B (1965), 73-77.

[3] Hakimi, S. L., "On Realizability of a Set of Integers as Degrees of the Vertices of a Linear Graph - I and II," J. Soc. Indust. Appl. Math., 10 (1962), 496-506 and 11 (1963), 135-147.

[4] Kleitman, D. J. and Wang, D. L., "Algorithms for Constructing Graphs and Digraphs with Given Valences and Factors," preprint (to be published), 1972.

[5] Kundu, S., "The k-factor conjecture is true," (to be published), 1971.

[6] Lehman, A., "A Solution to the Shanon Switching Game," J. Soc. Indust. Appl. Math., 12 (1964), 687-725.

[7] Rao, A. R. and Rao, S. B., "On factorable degree sequences," to appear in J. Comb. Theory.

[8] Rao, S. B., "Contributions to the Theory of Directed and Undirected Graphs," Ph.D. Thesis, 1970, Indian Statistical Institute.

[9] Senior, J. K., "Partitions and their representative graphs," Amer. J. Math., 73 (1951), 663-689.

UPPER AND LOWER BOUNDS ON THE PROBABILITY OF ERROR IN DIGITAL SYSTEMS

LOUIS E. JANKAUSKAS and ROBERT LUGANNANI
Department of Electrical Engineering
Princeton University, Princeton, N.J. 08540

ABSTRACT

In this paper we present a technique for finding upper and lower bounds on the probability of error in a digital system perturbed by intersymbol interference and additive Gaussian or non-Gaussian noise. The technique is valid for all noise densities that are monotone in two regions that depend upon the detector threshold and the maximum intersymbol interference. Examples are given for various noise densities and pulse shapes.

I. INTRODUCTION

Intersymbol interference and additive noise are major sources of degradation of high speed digital systems and finding the probability of error in such systems has long been a problem of interest [1-7]. The assumption of Gaussian noise is made in many systems because of the practical importance of this particular noise model and because of its mathematical tractability. However, the Gaussian assumption is not always valid and it is often necessary to consider other noise models such as impulsive noise caused by man-made electrical transients or atmospheric phenomena [8-9]. Most of the existing techniques for evaluating or bounding the probability of error have been developed for the Gaussian case and they are either not applicable or do not yield useful results in the non-Gaussian case. The purpose of this paper is to present a technique for obtaining upper and lower bounds for the probability of error that is valid for a large class of noise densities and does not rely upon the Gaussian assumption.

In this section we will describe the system. In the next section upper and lower bounds on the probability of error will be derived and conditions on the noise density for the bounds to be valid will be found. In section 3 we will present classes of noise densities for which the bounds are valid and examples using three types of received pulses will be given. Gaussian, Laplace, generalized Cauchy, and mixtures of these densities are used for the noise density in these examples.

Figure 1 is the block diagram of the system to be discussed. A sequence of independent, identically distributed, equiprobable binary numbers, $\{a_q\}$, is the information to be transmitted. This sequence is used to amplitude modulate a train of identical pulses, s(t), the transmitted pulse train being given by $g(t)=\Sigma\, a_q s(t-qT)$, where 1/T is the transmission rate. The channel and receiver filter are linear, time-invariant, and have overall impulse response h(t). We assume that the noise is additive and monotone in two regions that will be discussed in more detail below. The received pulse, r(t), is the response of the channel and receiver filter to an individual transmitted pulse, i.e.,

$$r(t)=\int_{-\infty}^{\infty} s(u)h(t-u)du \tag{1}$$

we can then express the sampler input as

$$y(t)=\sum_{-\infty}^{\infty} a_q\, r(t-qT)+n(t) \tag{2}$$

To simplify notation we will assume that $r(t_o)>0$ (this entails no loss of generality), where t_o is the sampling time. The input to the threshold detector of level K is given by

$$y(pT+t_o)=\sum_{-\infty}^{\infty} a_q\, r(t_o+pT-qT)+n(t_o+pT) \tag{3}$$

and the output sequence $\{b_p\}$ is obtained by the decision rule

$$b_p = \begin{cases} +1 & \text{if } y(pT+t_o)\geq K \\ -1 & \text{if } y(pT+t_o)<K \end{cases} \tag{4}$$

and since the $\{a_q\}$ are independent and identically distributed, the probability of error is the same for all p. Therefore, letting p=0 yields

$$y(t_o)=a_o r(t_o)+z(t_o)+n(t_o) \tag{5}$$

where $a_o r(t_o)$ is the signal term, $n(t_o)$ is the noise term and $z(t_o)=\Sigma' a_q\, r(t_o-qT)$ is the intersymbol interference term. The prime means that the q=o has been omitted in the sum.

Because of our assumption that $r(t_o)>0$, an error is made when $a_o=+1$ and $z(t_o)+n(t_o)<K-r(t_o)$ or when $a_o=-1$ and $z(t_o)+n(t_o)\geq r(t_o)+K$. Therefore, the probability of error is given by

$$\begin{aligned} P_e &= \tfrac{1}{2}\,\text{Prob.}\{n(t_o)<K-r(t_o)-z(t_o)\}+\tfrac{1}{2}\,\text{Prob.}\{n(t_o)\geq r(t_o)+K-z(t_o)\} \\ &= \tfrac{1}{2}P_B + \tfrac{1}{2}P_A \end{aligned} \tag{6}$$

Computational difficulties arise in the evaluation of P_e because, with the exception of a few special cases, the distribution of $z(t_o)$ is unknown.

II. DERIVATION OF BOUNDS

The upper and lower bounds on the probability of error will be derived in this section. The probability of error is given in Eq. 6 and, therefore, we must bound P_A and P_B. Since these two expressions have the same general form, bounds will be derived for P_A and then conditions on the noise density will be found when both forms are used. Theorem 1 will provide the lower bound and Theorem 2 will provide the upper bound.

We begin by defining the tail distribution

$$Q(x) = \int_x^{\infty} f(y)dy \tag{7}$$

where $f(x)$ is the noise probability density function. We say that $Q(x)$ satisfies Condition I when $Q(z+c)+Q(z-c)\geq Q(z+b)+Q(z-b)$ for all $|c|\geq|b|$.

Next, we will define three probabilities:

1) $P_A=\text{Prob.}\{n(t_o)\geq r(t_o)+K-z(t_o)\}$, where $z(t_o)$ is the intersymbol interference caused by the actual interferring pulses.

2) $P_L^{(N)}=\text{Prob.}\{n(t_o)\geq r(t_o)+K-z_L^{(N)}(t_o)\}$, where $z_L^{(N)}(t_o)$ is the intersymbol interference caused by a set of N interferring pulses.

3) $P_u^{(M)}=\text{Prob.}\{n(t_o)\geq r(t_o)+K-z_u^{(M)}(t_o)\}$, where $z_u^{(M)}(t_o)$ is the intersymbol interference which results when the absolute value of M interferring pulses have been added to form one interferring pulse and the remaining interferring pulses are unaltered.

Theorem 1

If Condition I holds, then $P_L^{(N)}\leq P_A$ for all N.

Proof: Select any N interferring pulses, then we have

$$P_L^{(N)}=\frac{1}{2^N}\sum_{j=1}^{2^N} Q(K+r(t_o)-z_{L,i}^{(N)}(t_o)) \tag{8}$$

where $z_{L,i}^{(N)}(t_o)$ are the 2^N possible values of $z_L^{(N)}(t_o)$. If we now add one interferring pulse to the pulse shape and find $P_L^{(N+1)}$, then we obtain

$$P_L^{(N+1)}=\frac{1}{2^N}\sum_{i=1}^{2^N}\frac{Q(K+r(t_o)-z_{L,i}^{(N)}(t_o)+u)+Q(K+r(t_o)-z_{L,i}^{(N)}(t_o)-u)}{2} \tag{9}$$

where u is the new interferring pulse. Letting $z=K+r(t_o)-z_{L,i}^{(N)}(t_o)$ in the hypothesis of the theorem, we obtain

$$P_L^{(N+1)}\geq P_L^{(N)}$$

furthermore, since $P_A=\lim_{N\to\infty} P_L^{(N)}$, we have $P_A\geq P_L^{(N)}$ for all N.

Theorem 2

If Condition I holds, then $P_A\leq P_u^{(M)}$ for all M.

Proof: The proof will be by induction on the number of pulses added. As a basis we will prove the theorem for M=2. Let a_j and a_k be the two interferring pulses whose absolute values are to be added and let $c=|a_j|+|a_k|$. Since $c\geq|a_j+a_k|$ and $c\geq|a_j-a_k|$, then from the hypothesis of the theorem it follows that

$$\frac{Q(z-c)+Q(z+c)}{2}\geq\frac{Q(z+a_j+a_k)+Q(z-a_j+a_k)+Q(z+a_j-a_k)+Q(z-a_j-a_k)}{4} \tag{10}$$

and from the definition of P_A

$$P_A=\lim_{N\to\infty}\frac{1}{2^N}\sum_{i=1}^{2^N}\frac{Q(r(t_o)+K-z_i'(t_o)+a_j+a_k)+Q(r(t_o)+K-z_i'(t_o)+a_j-a_k)}{4} + \frac{Q(r(t_o)+K-z_i'(t_o)\ a_j+a_k)+Q(r(t_o)+K-z_i'(t_o)-a_j-a_k)}{4} \tag{11}$$

where the $z_i'(t_o)$ are the 2^N possible values for N pulses that do not contain a_j or a_k. And by Eq. 10, letting $z=r(t_o)+K-z_i'(t_o)$, it follows that

$$P_A \le \lim_{N\to\infty} \frac{1}{2^N} \sum_{i=1}^{2^N} \frac{Q(z-c)+Q(z+c)}{2} \tag{12}$$

The sum in Eq. 12 is $P_u^{(2)}$ and the theorem holds for M=2. If we assume that the theorem is true for the sum of M interferring pulses, then we must show that it is also true for M+1. Let $a_1, a_2, a_3, \ldots$ be the interferring pulses and let $c=|a_1|+|a_2|+\ldots\ldots+|a_M|$. Therefore,

$$P_u^{(M)} = \lim_{N\to\infty} \frac{1}{2^N} \sum_{i=1}^{2^N} \frac{Q(K+r(t_o)-z_i''(t_o)+C)+Q(K+r(t_o)-z_i''(t_o)-C)}{2} \tag{13}$$

now let $cc=c+|a_{M+1}|$, then $cc\ge|c+a_{M+1}|$, $cc\ge|c-a_{M+1}|$ and we have

$$\frac{Q(z+cc)+Q(z-cc)}{2} \ge \frac{Q(z+c+a_{M+1})+Q(z+c-a_{M+1})+Q(z=c+a_{M+1})+Q(z-c-a_{M+1})}{4} \tag{14}$$

Using Eqs. 13,14 it follows that $P_u^{(M+1)} \ge P_u^{(M)}$. And by the inductive hypothesis $P_u^{(M)} \ge P_A$. Therefore, $P_u^{(M+1)} \ge P_A$.

Result 1

H(x), is a convex function on an interval I iff $H'(x_2)\ge H'(x_1)$ for all $x_2>x_1$ and x_1, x_2 in I.

Result 2

H(x) is a convex function on an interval I iff, for all $|c|\ge|b|$ and all z such that z+c and z-c and z-c are in I, we have

$$H(z+c)+H(z-c)\ge H(z+b)+H(z-b) \tag{15}$$

Proof: Define $G(x)=H(z+x)+H(z-x)$. Assume that H(x) is a convex function. Then we must show that G(x) is non-decreasing for $x\ge 0$. But from the definition of G(x) it follows that

$$G'(x)=H'(z+x)-H'(z-x) \tag{16}$$

and since H(x) is convex, it follows from result 1 that $G'(x)\ge 0$. Conversely, if $G'(x)\ge 0, x\ge 0$ it then follows that $H'(z+x)\ge H'(z-x)$ for all z and thus from result 1 it follows that H(x) is convex.

Conditions on the noise density that satisfy the hypothesis of Theorems 1 and 2 will be derived. By result 2 we must find the conditions on f(x) such that Q(x) is convex. Q(x) is convex iff $Q'(x)$ is non-decreasing. But $Q'(x)=-f(x)$ and, therefore, Q(x) is convex if f(x) is non-increasing. For $a_o=-1$, an error is made if $n(t_o)\ge r(t_o)+K-z(t_o)$, and since $z'(t_o)\le B(t_o) = \Sigma\, r(t_o-qT)$, then f(x) must be non-increasing in the interval $[r(t_o)+K-B(t_o), r(t_o)+K+B(t_o)]$. Similarly for P_B, we find that f(x) must be non-decreasing in the interval $[-r(t_o)+K-B(t_o), -r(t_o)+K+B(t_o)]$.

If the "eye" is closed, $B(t_o)>r(t_o)$, then these monotonicity conditions can only be met by a noise whose density function is flat in the interval $[r(t_o)+K-B(t_o), -r(t_o)+K+B(t_o)]$.

In summary, Theorem 1 states that the truncated pulse shape yields a lower bound for all noises with a distribution that satisfies Condition I. Since the lower bound is the greatest for the largest interferring pulses, then instead of simply truncating the pulse train, one should select the largest interferring pulses to obtain the tightest bound. Theorem 2 states that the absolute value of a countable number of pulses can be added to obtain an upper bound on P_A. One can continue to combine the interferring pulses in this manner until there are only 2N interferring pulses. The resulting Prob.$\{n(t_o) \geq r(t_o)+K-z(t_o)\}$ will then be an upper bound on P_A.

III. EXAMPLES

Examples will be presented to illustrate the bounds and to show the effects of non-Gaussian noise. These examples use three pulse shapes. The constants in the three pulse shapes were chosen such that a close fit to the $\sin(|t|/_T)/|t|/T$ pulse was obtained for $|t| \leq T/2$.

Gaussian Pulse

The Gaussian pulse decays very rapidly and is given by

$$r(t)=\mathrm{EXP}[-(8t/5T)^2] \tag{17}$$

Because of the rapid decay of this pulse shape, the convergence of the upper and lower bounds is very fast.

Fourth-Order Chebyshev Pulse

An example of a pulse shape that decays moderately is the Fourth-Order Chebyshev Pulse and is given by

$$r(t)=A_1\cos(w_1|t|/T-\phi_1)\mathrm{EXP}[-x_1|t|/T]+A_2\cos(w_2|t|/T-\phi_2)\mathrm{EXP}[-x_2|t|/T] \tag{18}$$

where the constants are given in Ref [4]. In the calculation of the upper bound 200 pulses were combined to form 14 pulses. The heuristic of adding the next generated interferring pulse to the smallest of the sums of pulses was used.

Sine-Squared Pulse

An example of a slowly decaying pulse is the $\sin^2(.72\ t/T)/(.72\ t/T)^2$ pulse. In the examples effects of 1000 pulses were used to form 10 pulses.

A sufficient condition for the bounds of section 2 to be valid is that the noise probability density function be non-increasing in the interval $[r(t_o)+K-B(t_o), r(t_o)+K+B(t_o)]$ and non-decreasing in the interval $[-r(t_o)+K-B(t_o), -r(t_o)+K+B(t_o)]$. Two classes of noise densities that are increasing on the interval $(-\infty,0)$ and decreasing on the interval $(0,\infty)$ will be discussed. These classes are generalizations of the Gaussian and Cauchy distributions. Mixtures of these densities will also be discussed.

Generalized Gaussian noise [9]

This class of noise densities is a generalization of the Gaussian and Laplace densities. The generalized Gaussian density is even, unimodal and is given by

$$f(x)=\frac{c\eta(t_\sigma, c)}{2\Gamma(1/c)}\exp\left[-(\eta(\sigma_n, c)|x|)^c\right] \tag{19}$$

where we have defined

$$\eta(\sigma,c)= \frac{1}{T}\left[\frac{\Gamma(3/c)}{\Gamma(1/c)}\right]^{\frac{1}{2}} \tag{20}$$

in the above equations, c is a positive parameter, $\Gamma(\cdot)$ is the gamma function, and σ_n^2 is the noise variance.

Generalized Cauchy noise [8,9]

The generalized Cauchy class of noise densities takes the form

$$f(x)=\frac{c\eta(\sigma,c)\Gamma(v+{}^{1}/c)}{2\Gamma(1/c)\;\Gamma(v)\;v^{1/c}}\left\{1+\frac{[\eta(\sigma,c)|x|]^{c}}{v}\right\}^{-(v+{}^{1}/c)} \tag{21}$$

where c and v are positive parameters. The Cauchy density is contained in this class. The variance of generalized Cauchy noise is [9]

$$\sigma_n^{2}= \frac{\sigma^2 v^{2/c}\,\Gamma(v-2/c)}{\Gamma(v)} \tag{22}$$

and this is only finite for $cv>2$. As v goes to infinity σ_n^2 approaches σ^2. Generalized Cauchy noise can be used as a model for atmospheric noise [8].

Mixture densities

In many practical cases the non-Gaussian noises discussed in the previous paragraphs are not an accurate discription of the noise. A digital system in which the noise has a Gaussian part and an impulsive part does not fit the previous classes, but can be modeled as a mixture density. That is, the noise density can be modeled as

$$f(x)=\epsilon f_1(x) + (1-\epsilon)f_2(x) \tag{23}$$

where ϵ is a constant between 0 and 1. $f_1(x)$ and $f_2(x)$ are the densities that are to be mixed. If $f_2(x)$ is Gaussian and $f_1(x)$ is Laplace or Cauchy, then f(x) could be a model for impulse noise. The probability of error for mixture densities is easy to find,

$$P_e= \epsilon P_e'+(1-\epsilon)P_e'' \tag{24}$$

where P_e' and P_e'' are the probabilities of error associated with $f_1(x)$ and $f_2(x)$, respectively.

Figure 2 presents the convergence of the bounds as the number of interferring pulses formed increases. This is presented for the three pulse shapes and Gaussian and Laplace noise. Figures 3,4, and 5 show the sensitivity of the three pulse shapes to sampling time offset. In these examples the Cauchy and the generalized Cauchy noises are given by Eq. 29 with $v=1/2, c=2$ and $v=2, c=2$ respectively. The variance of the Gaussian and Laplace are chosen such that

$$\sigma_L^2 = \sigma_G^2 = \sigma^2 \tag{25}$$

where σ^2 is the parameter in Eqs. 21,22.

Figure 6 exhibits the sensitivity of the Chebyshev pulse to sampling time offset for a Gaussian and Laplace mixture.

This is shown for various $\gamma^2=\sigma_L^2/\sigma_G^2$. In figures 7,8 we show how the probability of error varies as a function of γ^2 for various ϵ. Gaussian and Laplace mixtures are presented in figure 7. Gaussian and generalized Cauchy mixtures are presented in figure 8.

Conclusions that can be drawn from the examples are: (1) For noise densities of equal variance Gaussian noise has the lowest probability of error. In the cases considered the Gaussian noise probability of error was several orders of magnitude lower than the Laplace or the generalized Cauchy probability of error. (2) The probability of error in Gaussian noise is the most sensitive to sampling time offset. (3) If one considers impulse noise to be represented by Laplace or generalized Cauchy noise then (figures 7,8), impulse noise seriously degrades the system. Even in moderate impulse power and low probability of impulse noise ($\gamma^2=10, \epsilon=.001$) the probability of error was 4 orders of magnitude above the Gaussian noise case.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of the National Science Foundation under Grant GK-24187 and of the U.S. Army Research Office under Contract DAHCO4-69-0012.

REFERENCES

1) M.R. Aaron and D.W. Tufts, "Intersymbol interference and error probability", IEEE Trans. Inform. Theory, Vol.IT-12, Jan. 1966, pp. 26-34.

2) E.Y. Ho and Y.S. Yeh, "A new approach for evaluating the error probability in the presence of intersymbol interference and additive Gaussian noise", BSTJ, Vol. 49, Nov. 1970, pp. 2249-2265.

3) F.S. Hill, "The computation of error probability for digital transmission", BSTJ, Vol. 50, July-Aug. 1971, pp. 2055-2077.

4) R. Lugannani, "Intersymbol interference and probability of error in digital systems", IEEE Trans. Inform. Theory, Vol. IT-15, Nov. 1969, pp. 682-688.

5) B.R. Saltzberg, "Intersymbol interference error bounds with application to ideal bandlimited signaling", IEEE Trans. Inform. Theory, Vol. IT-14, July 1968, pp. 563-568.

6) O. Shimo and M.I. Celebiler, "The probability of error due to intersymbol interference and Gaussian noise in digital communication systems", IEEE Comm. Tech., Vol. Com-19, April 1971, pp. 113-119.

7) Y.S. Yeh and E.Y. Ho, "Improved intersymbol interference error bounds in digital systems", BSTJ, Vol. 50, Oct. 1971 pp. 2585-2598.

8) H.M. Hall, "A new model for impulsive phenomena: Application to atmospheric-noise communication channels", Stanford Electron Laboratory, Stanford University, Tech. Rep. 3412-8 and 7050-7, April 1966.

9) J.H. Miller and J.B. Thomas, "Detectors for discrete-time signals in non-Gaussian noise", IEEE Trans. Inform. Theory, Vol. IT-18, March 1972, pp. 241-250.

10) T.S. Ferguson, Mathematical Statistics, New York: Academic Press, 1967.

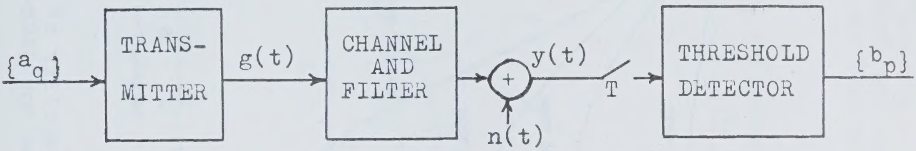

{a_q}
TRANS-MITTER
g(t)
CHANNEL AND FILTER
+
y(t)
n(t)
T
THRESHOLD DETECTOR
{b_p}

Fig. 1 Block Diagram

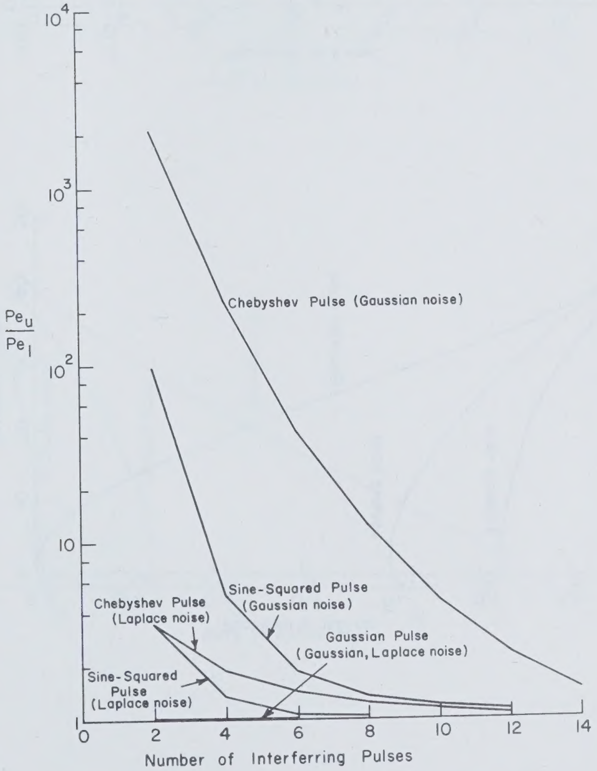


Fig. 2 Convergence of Bounds

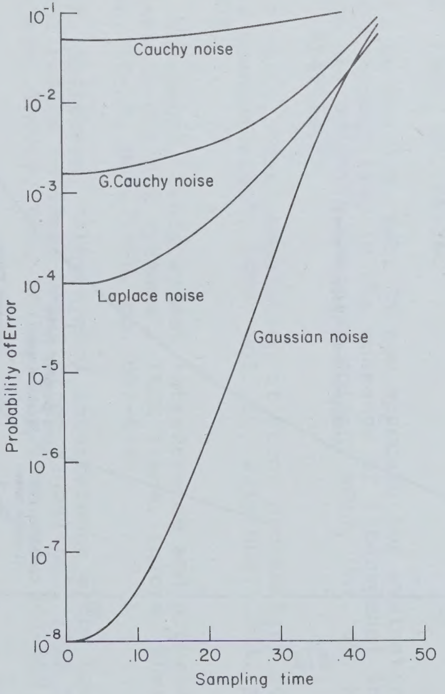


Fig. 3 Probability of Error versus Sampling Time for Gaussian Pulse

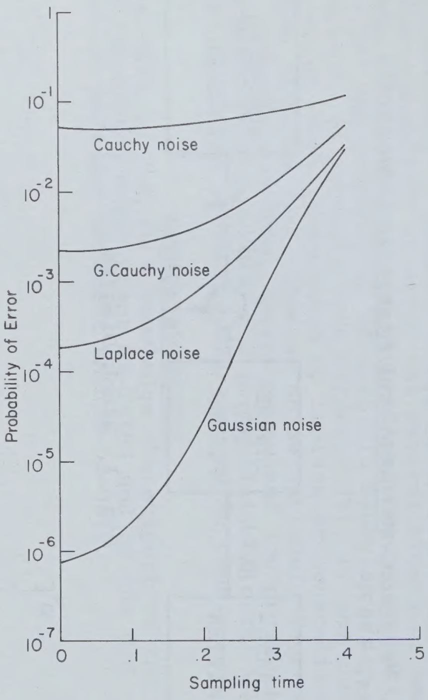


Fig. 4 Probability of Error versus Sampling Time for SINE-Squared Pulse

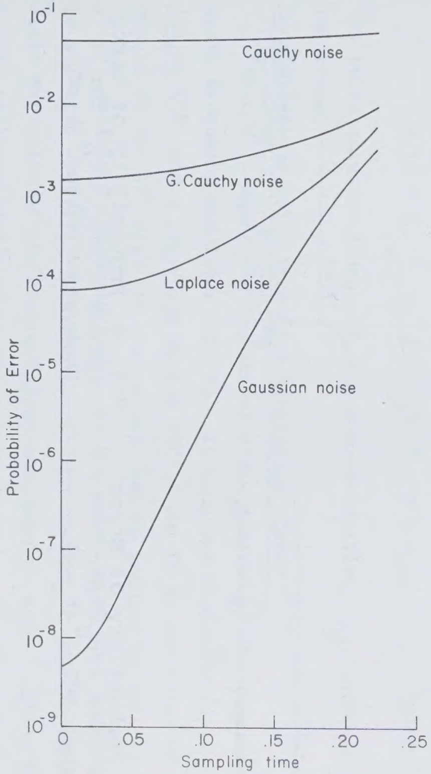


Fig. 5 Probability of Error versus Sampling Time for Chebyshev Pulse

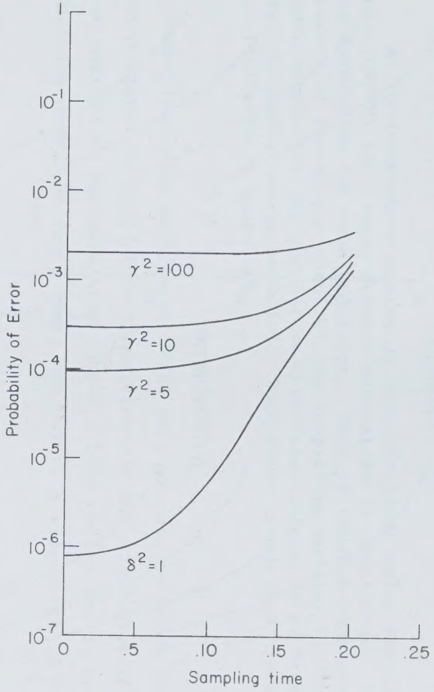


Fig. 6 Probability of Error versus Sampling Time for Chebyshev Pulse with $\epsilon=.01, \gamma^2=\sigma_L{}^2/\sigma_G{}^2$.

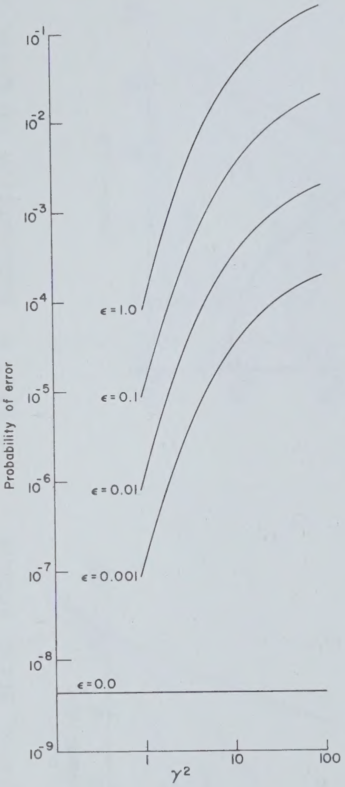


Fig. 7 Probability of Error Versus $\gamma^2=\sigma_L^2/\sigma_G^2$ for Chebyshev Pulse (t_o=0).

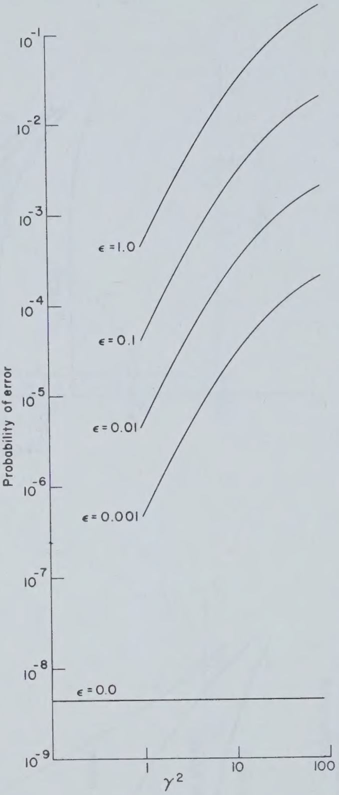


Fig. 8 Probability of Error Versus $\gamma^2=\sigma_{GC}^2/\sigma_G^2$ for Chebyshev Pulse (t_o=0).

FINITE-PULSE STABILITY OF INTERCONNECTED SYSTEMS WITH COMPLETE-RESET PULSE FREQUENCY MODULATORS

HALİL Ö. GÜLÇÜR
NATO-FELLOW, DEPT. OF ELEC. ENG'G.
NEWARK COLLEGE OF ENGINEERING
NEWARK, NEW JERSEY

ANDREW U. MEYER
PROFESSOR OF ELEC. ENG'G.
NEWARK COLLEGE OF ENGINEERING
NEWARK, NEW JERSEY

ABSTRACT

An interconnected system of complete-reset pulse frequency modulators (CRPFM) and linear dynamic subsystems is considered. Upper bounds are determined for each modulator. Each of these bounds represent a measure of the energy spent by the respective modulator. Finiteness of these bounds for all modulators consitutes Finite-Pulse Stability and implies finite energy expanded. Sufficient conditions are established for Finite-Pulse Stability.

INTRODUCTION

Pulse frequency modulation (PFM) is a process by which information of an input signal is coded into time-intervals and polarities of identical pulses emitted as the modulator output. This coding can be accomplished in an infinite number of ways; thus, PFM covers a wide class of pulse modulation.

The output signal of a pulse frequency modulator, u(t), is defined in terms of a sequence of impulses of equal strength M and of impulse-polarity $\varepsilon_i = \mp 1$, emitted at time-instances t_i $(i = 1,2,\ldots)$, i.e.,

$$u(t) = M \sum_{k=1}^{N} \varepsilon_k \, \delta\,(t-t_k); \; 0 < t < t_{N+1} \tag{1}$$

The pulse emission times, t_k, and pulse-polarities, ε_k, follow some functional relations.

Complete-Reset Pulse Frequency Modulation (CRPRM)[1]

Figure 1. shows the functional block diagram of the complete-reset pulse frequency modulator (CRPFM). It consists of a resettable timing filter (TF) and a threshold device (TD). The TD is activated by the output signal of the TF, z(t), in such a way that an impulse is emitted whenever $|z(t)|$ exceeds a threshold-level, S; the polarity of that impulse is equal to the sign of z(t). Furthermore, at the instant of impulse-emission all state variables of the timing filter are reset to zero.[2] Thus, the functional relations defining t_k and ε_k are given by

---

[1] CRPFM is a generalization of the Integral Pulse frequency modulator (IPFM) [1], [2], [3] and incorporates the features of the "nth order neural trigger" of Pavlidis [28], the "functional pulse frequency modulator" of Jury and Blanchard [9], the "type II pulse modulator" of Skoog and Blankenship [17] and the "pulse frequency modulator of the second kind with complete clearing of the time-marking filter" of Kuntsevich and Chekhovoi [13].

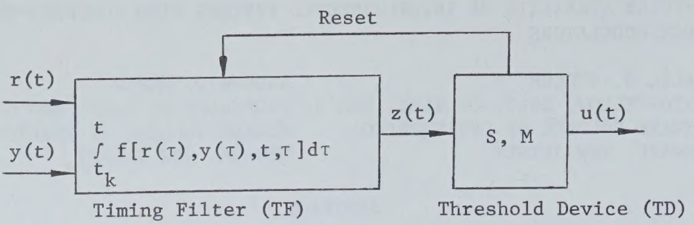


Fig. 1. Block diagram representation of the Complete Reset Pulse Frequency Modulator (CRPFM)

$$z(t) = \int_{t_{k-1}}^{t} f[r(\tau),y(\tau),t,\tau]d\tau; \; t_{k-1} < t < t_k \tag{2}$$

$$t_k = \min \{t \mid t > t_{k-1} \text{ and } |z(t)| \geqslant S\} \tag{3}$$

$$\varepsilon_k = \operatorname{sgn} [z(t_k^-)] \tag{4}$$

where r(t) and y(t) represent two modulator-input signals; r(t) represents an external input and y(t) could be a feedback signal from a linear element in the overall system containing this CRPFM.

The function $f[r(\tau),y(\tau),t,\tau]$ is usually of the form

$$f[r(\tau),y(\tau),t,\tau] = g_0(t,\tau)\; e(\tau) \tag{5.a}$$

$$e(t) = f_0[r(t),y(t),t] \tag{5.b}$$

where $f_0[\;]$ is a memoryless,time-varying nonlinearity and the kernel $g_0(t,\tau)$ is the impulse response of a (usually RC lowpass) single-input single output linear dynamic system. For IPFM, it is $g_0(t,\tau)=1$ and for relaxation pulse-frequency modulation (RPFM) [2], it is $g_0(t,\tau)=e^{-\frac{t-\tau}{T_R}}$.

Interconnected System Consisting of CRPFM's and Linear Dynamic Elements

The system is illustrated in Fig. 2. The PFM block contains $\underline{m}$ CRPFM's (m=1,2,..). The modulator output vector, $\underline{u}(t)$, is applied to a linear dynamical subsystem (LP) whose impulse response matrix is $\underline{G}(t,\tau)$. A combination of the output vector of the Linear Part (LP), $\underline{y}(t)$, and an external input vector, $\underline{r}(t)$, is fed to the modulator block. $\underline{y}_0(t)$ is the initial condition response vector of the LP which could also include disturbances.

---

[2] A variation of CRPFM, which is closer to PFM in the nervous system, uses a TD that emits an impulse when the TF-output signal Z(t) (instead of its absolute value |Z(t)|) exceeds a threshold level. This version is called Single-Signed CRPFM.

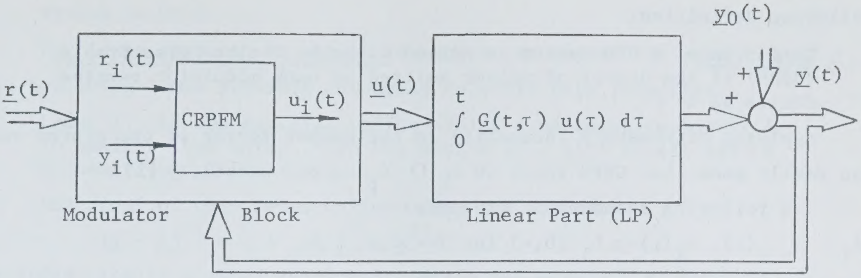


Fig. 2. Block diagram of an interconnected system consisting of CRPFM's and linear dynamical elements.

Let $t_{i,j}$ be the instant at which the ith modulator emits its jth pulse and let $K_i(t)$ denote the total number of pulses emitted by the ith modulator prior to time t. Then the operation of the system is given by the relation for the ith timing-filter output signal (leading to the ith TD):

$$z_i(t) = \int_{t_{i,K_1(t)}}^{t} f_i\{r_i(\tau), y_{0,i}(\tau) + \sum_{j=1}^{m} M_j \sum_{k=1}^{K_j(t)} \operatorname{sgn}[z_j(t^-_{j,k})]\, g_{i,j}(\tau, t_{j,k}),\ t, \tau\} d\tau$$

$$(i = 1,2,..,m) \qquad (6.a)$$

Consider a fixed time t^*. After time t^*, the next impulse of the system may be emitted by any one of the m modulators. Let t^f be the firing time of the next impulse; it is given by

$$t^f = \min_{i=1,..,m} \{t \mid t > t_{i,K_i(t^*)} \text{ and } |z_i(t)| \geqslant S_i\} \qquad (6.b)$$

The identification number of the modulator is

$$\ell = \{i = 1,2,..,m \mid |z_i(t^f)| \geqslant S_i\} \qquad (6.c)$$

Thus,

$$t_{\ell,K_\ell(t^*)+1} = t^f \qquad (6.d)$$

Equations (6) are the basic equations defining the PFM system.

GLOBAL FINITE-PULSE STABILITY IN PFM SYSTEMS

Stability in PFM systems has been the subject of several investigations. The tools used in most of these works were Lyapunov's second method [5]-[13] and frequency response criteria (Popov's method) [14] - [20]. Some authors have obtained criteria by direct application of basic functional properties of the system [17], [20], [21], [22]. This is the approach employed here. The number of pulses emitted from a modulator is a measure of the energy spent by that modulator during the operation of the system. Thus the stability of a PFM system can be related to this variable, which leads to the following definition:

> Definition: A PFM system is called Globally Finite-Pulse Stable (GFPS) if the number of pulses emitted by each modulator remains finite as $t \to \infty$.

Applying Minkowski's inequality to the output vector of the system one can easily show that GFPS leads to $L_1 \cap L_p$ output stability [23].

The following assumptions are made:

(A_1) $y_{o,i}(t)$, $r_i(t) \ \varepsilon \ L_1 \ [0,\infty)$ for $i = 1,2,..,m$

(A_2) There exist functions $g'_{i,j}(t) \ \varepsilon \ L_1 \ [0,\infty)$ such that $\forall t>\tau$; $t,\tau \ \varepsilon \ [0,\infty)$ and $i,j = 1,2,..,m$

$$|g_{i,j}(t,\tau)| \leqslant |g'_{i,j}(t-\tau)|$$

(A_3) There exist finite nonnegative constants α_i and β_i, such that $\forall t \ \varepsilon \ [0,\infty)$ and $i = 1,2,..,m$

$$|f_i \ [r_i, \ y_i, \ t,\tau] \ | \leqslant \alpha_i |r_i| + \beta_i |y_i|$$

Using the inequality $|\int f(t)dt| \leqslant \int |f(t)|dt$ together with assumption A_3 in equations (6) and applying the triangle inequality $|A|+|B| \geqslant |A+B|$, an inequality for each firing instant $t_{i,v} (v=1,2,..,K_i)$ is obtained. Summing these inequalities for $v=1,2,..,K_i$ and making use of the causality of the linear part of the system and using assumption A_2 and then changing the dummy variable τ to $\tau+t_{j,k}$ and replacing $t_{i,k_i}-t_{j,k}$ by t, the following matrix inequality is obtained:

$$[\underline{I} - \underline{H}'(t)] \ \underline{k}(t) \leq \underline{v}(t) \tag{7}$$

In eq. (7), $\underline{k}(t)$ is the m-dimensional vector formed by the number of firings of the modulators prior to time t, $\underline{v}(t)$ is an m-dimensional column vector whose ith element is defined by

$$v_i(t) = \int_0^t \{[\alpha_i |r_i(\tau)| + \beta_i |y_{0,i}(\tau)|]/S_i\} \, d\tau \tag{7.a}$$

and $\underline{H}'(t)$ is an m-dimensional square matrix whose element in the ith row and jth column is

$$h'_{i,j}(t) = \beta_i M_j \int_0^t [|g'_{i,j}(\tau)|/S_i] \, d\tau \tag{7.b}$$

When the matrix $[\underline{I} - \underline{H}'(t)]^{-1}$ exists and is nonnegative [24] - [26], inequality (7) can be transformed into

$$\underline{k}(t) \leqslant \mathrm{In} \ \{[\underline{I} - \underline{H}'(t)]^{-1}\underline{v}(t)\} \tag{8}$$

where, the notation In { } stands for the integer part of the corresponding vector. Inequality (8) determines the upper bounds of the number of pulse emissions as $t \to \infty$.

Stated briefly, the above results, together with the above definition, lead to the following theorem:

> Theorem: If assumptions A_1 to A_3 of this section are satisfied and if the matrix $[I - \underline{H}'_\infty]^{-1}$ is nonnegative, then the CRPFM

system is GFPS.

For large systems, it may be cumbersome to invert the matrix $[\underline{I}-\underline{H}'_\infty]$. The following lemma provides the means to avoid this inversion.

Lemma 1: If the spectral radius $\lambda(\underline{A})$ of a nonnegative matrix $\underline{A}$ is smaller than unity, then the matrix $[\underline{I} - \underline{A}]^{-1}$ exists and is nonnegative.

The proof of this lemma follows from the identity:

$$[\underline{I} - \underline{A}]^{-1} = \underline{I} + \underline{A} + \underline{A}^2 + \underline{A}^3 + \ldots$$

provided, $\lambda(\underline{A}) < 1$ ([27], page 60). Since the matrix $\underline{A}$ is nonnegative, so is $\underline{A}^2$, $\underline{A}^3$,... and their sum; hence the proof.

Lemma 1, together with the Theorem of this section, gives the following corollary:

Corollary 1: If assumptions A_1 to A_3 are satisfied and if $\lambda(\underline{H}'_\infty) < 1$, then the CRPFM system is GFPS.

With present computer technology, it is not difficult to calculate the eigenvalues of a matrix. However, the following lemma, in most cases, eliminates this need.

Lemma 2: If $\underline{A}$ is an mxm nonnegative matrix and if

$$\sum_{j=1}^{m} a_{ij} < 1 \quad ; \quad i = 1,2,..,m$$

then the matrix $[\underline{I} - \underline{A}]^{-1}$ is also nonnegative.

The proof of this lemma follows from the Gersgorin's Theorem [25] and Lemma 1.

Corollary 1, Lemma 2. and the Theorem of this section lead to the following corollary:

Corollary 2: If assumptions A_1 to A_3 are satisfied and if

$$\lim_{t\to\infty} \frac{\beta_i}{S_i} \sum_{j=1}^{m} M_j\, h_{i,j}(t) < 1$$

then the CRPFM system is GFPS.

For example, when applied to a single-loop, single RPFM (or IPFM) system with a time-invariant linear element, the Theorem imposes the condition

$$\frac{M}{S}\, ||g||_1 < 1 \tag{9}$$

where, $||g||_1 = \int_0^\infty |g(t)|dt$. If there are two RPFM's (or IPFM's) in the (single) loop, (i.e., $g_{ii}(t) = 0$, i=1,2), then the condition becomes

$$\frac{M_1 M_2}{S_1 S_2}\, \|g_{12}\|_1 \cdot \|g_{21}\|_1 < 1 \tag{10}$$

Inequality (9) was obtained independently and almost simultaneously by Skoog and Blankenship [17], Kan [18] and (10) by Guy [21].

It is important to note that, although Corollary 2 is easier to apply than Corollary 1 (which, in turn, is easier to apply than the Theorem), they are not equivalent. Therefore, it is recommended to use Corollary 2 first and, if it fails, to refer to Corollary 1 and then to the Theorem.

CONCLUSIONS

An interconnected system consisting of a finite number of stable linear elements and finite number of Complete-Reset Pulse frequency modulators (CRPFM) is considered. Upper bounds on the number of pulses emitted by each modulator during the operation of the system are determined. Such number is indicative of the amount of energy spent by the corresponding modulator and thus the upper bound of the number of pulses emitted by all modulators represents a measure of stability. Sufficient conditions under which this number is finite are established and are shown to depend on nonnegativity of a certain matrix.

The features of the results are:

(1) Generality: The conditions apply to PFM systems containing distributed and/or lumped linear parts; the timing filters (TF's) are allowed to include nonlinearities. The number of modulators and/or the number of loops are not limited to one; the modulators are quite general (not restricted to IPFM or RPFM) and can be single-signed or double-signed.

(2) Simplicity: Once the $\underline{H}'_{\infty}$ matrix is known, it is relatively easy to apply the stability conditions.

Though the results apply to interconnected systems containing an arbitrary number of modulators, the application to a single-loop single-modulator system yields a condition which was previously obtained [17], [22] and examples (not presented here for lack of space) yield stability regions comparable (often better) to those obtained by other methods (such as described in [5]-[20]).

The present results require all linear plants to be asymptotically stable and all input signals to be absolutely integrable. In some cases where the linear plants contain integration and/or the input signals contain d-c parts, global finite pulse stability may still exist; the establishment of conditions for these cases are left for future investigation.

ACKNOWLEDGEMENT

The NATO fellowship of H. Ö. Gülçür is administered by the Scientific and Technical Research Council of Turkey (TUBITAK).

REFERENCES

[1] Jones, R.W., C.C. Li, A.U. Meyer and R.B. Pinter, "Pulse Frequency Modulation in Physiological Systems" IRE Trans. Biomed. Electronics, v. BME-8, n.1., Jan. 1961, pp. 59-67.

[2] Meyer, A.U., "Pulse Frequency Modulation and Its Effects in Feedback Control Systems" Ph.D. Dissertation, Northwestern Univ., Evanston, Illinois, Aug. 1961.

[3] Li, C.C., "Integral Pulse Frequency Modulated Control Systems" Ph.D. Dissertation, Northwestern Univ. Evanston, Illinois, June 1961.

[4] Pavlidis, T. and E.I. Jury, "Analysis of a New Class of Pulse Frequency Modulation Feedback Systems" IEEE Trans. AC, v. 10, Jan. 1965, pp. 35-43.

[5] Farrenkopf, R.L., A.E. Sabroff and P.C. Wheeler, "Integral Pulse Frequency On-Off Attitude Control System" Space Technology Laboratories, Inc., Report no. 8637-6117-RU000, 12 March 1963.

[6] Pavlidis, T., "Stability of a Class of Discontinuous Dynamical Systems" Information and Control, v.9, pp. 298-322, 1966.

[7] Clark, J.P.C., "Stability of a Class of Pulse Frequency Modulation Closed Loop Control Systems" IEEE Int. Conv. Rec., 1966, v.14, Pt 6, pp. 179-185.

[8] Kuntsevich, V.M. and Yu. N. Chekhovoi, "Investigating Stability of Sampled Data Control Systems with Pulse Frequency Modulation Using Lyapunov's Direct Method" Automation and Remote Control, n.2, Feb 1967, pp. 80-92.

[9] Jury, E.I. and J. Blanchard, "A Nonlinear Discrete System Equivalence of Integral Pulse Frequency Modulated Systems" JACC, June 1967, Philadelphia, Pa., pp. 680-693.

[10] King-Smith, E.A. and R.J. Cumpston, "The Stability of IPFM Systems" Int. Journal of Control, v.7, no.4, 1968, pp. 301-316.

[11] Shortle, G.E., Jr. and F.J. Alexandro, Jr., "Stability of Class of Pulse Frequency Modulated Control Systems" 1969 JACC, Boulder, Colo.

[12] Varadarajan, M.S., "Stability of Feedback Systems with Pulse-Frequency Modulation" Int. J. Control, V.13, n.2., pp. 265-272, 1971.

[13] Kunsevich, M. and Yu, N. Chekhovoi, "Asymptotic Stability on the Whole of One Class of Pulse-Frequency Systems of Second Type" Avtomatika i Telemekhanika, n.3., pp. 47-56, March 1971.

[14] Dymkov, V.I., "Absolute Stability of Pulse Frequency Systems" Automation and Remote Control, n. 10, Oct. 1967, pp. 109-114.

[15] Gelig, A., Kh., "Absolute Stability of Nonlinear Pulse Systems and Width and Time Modulation" Avtomatika i Telemekhanika, 29, n.7, pp.33-43, July 1968.

[16] Wylie, B.K. and R.V. Monopoli, "Stability Analysis of a Class of PFM Error Sampled Control Systems" Proc. of the 13th Midwest Symposium on Circuit Theory, May 7-8, 1970., Univ. of Minnesota, Minneapolis.

[17] Skoog, R.A. and G.L. Blankenship, "Generalized Pulse-Modulated Feedback Systems: Norms, Gains, Lipschitz, Constants and Stability" IEEE Trans. AC, v. 15, June 1970, pp 300-315.

[18] Kan, E.P.F. and E.I. Jury, "On Popov Criteria for Pulse Frequency Modulated Systems" Int. J. Control, v.13, n.6, pp. 1121-1129, 1971.

[19] Yu. N. Chekhovoi, "Stability in the Large of a Pulse-Frequency Modulation of Second Type" Avtomatika i Telemekhanika, n.11, pp. 53-60, Nov. 1971.

[20] Gelig, A. Kh., "Stability of Sampled Data Systems with Pulse-Frequency Modulation of Second Type" Avtomatika i Telemekhanika, n.11, pp. 53-60, Nov. 1971.

[21] Guy, Warren J., "Oscillations in PFM Control Systems" Ph.D. Dissertation, Newark College of Eng., Newark, N. J., 1970.

[22] Kan, E.P.F., "L_1 L_p Stability of an IPFM System" Int. J. Control, v.14, n.1, pp. 27-31, 1971.

[23] Willems, J.C., "The Analysis of Feedback Systems", The M.I.T. Press, Research Monograph No. 62, Cambridge, Massachusetts, 1971.

[24] Gantmacher, F.R., "Applications of the Theory of Matrices" Interscience Publishers, Inc., New York, 1959.

[25] Lancaster, P., "Theory of Matrices" Acad. Press, 1969.

[26] Bellman, R., "Introduction to Matrix Analysis" McGraw-Hill, 1960.

[27] Barnett, S. and Storey, "Matrix Methods in Stability Theory" Barnes & Noble, Inc., N.Y., 1970.

[28] Pavlidis, T., "A New Model for Simple Neural Nets and Its Application in the Design of a Neural Oscillator" Bulletin of Mathematical Biophysics, v.27, pp. 215-229, 1965.

THE ADVANTAGES OF COMBINING PHASE AND INJECTION LOCK

MELVIN SANDLER
The Cooper Union
Cooper Square
New York, N. Y. 10003

ABSTRACT

The nonlinear phase shift properties of an injection locked oscillator offer many interesting design possibilities. However, discussion is limited to those applications which combine the advantages of both phase and injection lock techniques. The result is conceptually new hardware solutions to such classical problems as occur in phase comparisons and FM threshold extension.

INTRODUCTION

Consider an oscillator, pulled from rest frequency (f_{rest}), to precisely the frequency of an injected signal (f_i). The oscillator, now synchronized by the injected signal is shifted in phase by an angle (α) dependent upon the frequency difference ($\Delta f = f_i - f_{rest}$) between the two states of oscillation (equation 1).[1,2]

$$\sin \alpha = 2Q \frac{E_O}{E_S} \frac{\Delta f}{f_{rest}}, \quad \frac{E_S}{E_O} << 1 \tag{1}$$

E_O, E_S are the oscillator and injected signal amplitudes at the point of injection.

Q is the loaded Q of the oscillator tank circuit

A variable phase shift is produced by electronically or mechanically varying the unsynchronized frequency (f_{rest}). Thereby, the injection locked oscillator becomes a variable phase shifter, which may be voltage controlled ($\Delta f = -ke$), or mechanically tuned. Since the injected signal is much weaker than the oscillator output, the oscillator simultaneously provides a stage of gain.

PHASE DETECTION

Figure 1 shows an injection locked oscillator combined with a phase control feedback loop. The steady state output of the phase detector is given by:

$$e = \gamma E_O E_S \sin(\theta + \alpha) \tag{2}$$

where θ is the phase difference to be detected

γ is the conventional phase detector constant.

Combining equations (1) and (2) with $\Delta f = -ke$ results in:

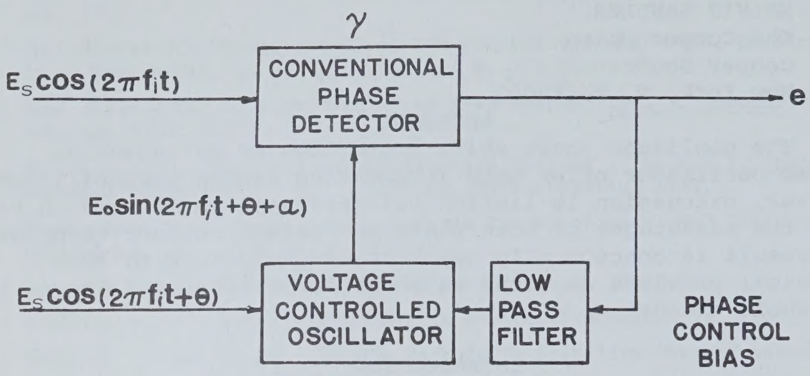

γ
$E_s\cos(2\pi f_i t)$
CONVENTIONAL PHASE DETECTOR
e
$E_o\sin(2\pi f_i t+\theta+\alpha)$
$E_s\cos(2\pi f_i t+\theta)$
VOLTAGE CONTROLLED OSCILLATOR
LOW PASS FILTER
PHASE CONTROL BIAS

FIGURE 1. INJECTION LOCKED PHASE DETECTOR

$$\sin^{-1}\frac{e}{\gamma E_o E_s} = \theta - \sin^{-1}2Q\frac{E_o}{E_s}\,\frac{ke}{f_{rest}} \qquad (3)$$

and if the condition $2QE_o^2k\gamma = f_{rest}$ is met:

$$e = \gamma E_o E_s \sin\left(\frac{\theta}{2}\right) \qquad (4)$$

with errors on the order of $\frac{\Delta f}{f_{rest}} <<< 1$

Equation (4) is a statement of unambiguous phase detection with a linear range double that of a conventional phase detector. It is easily shown that if n injection locked oscillators are injection locked in tandem (and controlled by the same bias voltage (e)):

$$e = \gamma E_o E_s \sin\left(\frac{\theta}{n+1}\right) \qquad (5)$$

Inspection of equation (5) reveals that as n is increased, the full 360 degree range of θ can be obtained over a decreasingly smaller portion of the sine curve.

Two nonlinear relations govern the operation of the extended range phase detector. The differential equation describing the behavior of the phase control loop is

$$\frac{2QE_ok}{f_{rest}E_{s2}}\cdot e = \sin\delta_n + \tau\frac{d}{dt}\sin\delta_n \qquad (6)$$

where $\delta_n = \frac{1}{n}\sin^{-1}\frac{e}{\gamma E_o E_{s1}} - \theta$

τ is the time constant of the low pass filter and E_{s1} does not necessarily equal E_{s2}

For phase differences $\frac{|\theta|}{n+1} < 60$ degrees the time constant approaches that of a linerized circuit. (i.e. the loop time constant (τ_L) approaches $\tau_L = \dfrac{\tau}{1+n\frac{E_{s2}}{E_{s1}}}$ for any value of n.) (7)

In addition to the phase lock transient inherent in the operation of the extended range phase detector, each oscillator exhibits an injection lock transient. Given the conditions that the oscillator output is much stronger than injected synchronizing signal and that the difference frequency is a small percentage of the oscillator rest frequency (f_{rest}), Adler[1] has shown that

$$\frac{d\alpha}{dt} = \frac{-E_s}{E_o}\frac{\pi f_{rest}}{Q}\sin\alpha + 2\pi\Delta f \tag{8}$$

After synchronization of the oscillator to a steady state impressed signal the condition of injection lock stated in equation (1) is met. i.e.

$$\alpha = \sin^{-1} 2Q\frac{E_o}{E_s}\frac{\Delta f}{f_{rest}}$$

Since sin α is constrained between +1 and -1, the condition for synchronization may be written as:

$$2Q\frac{\Delta f}{f_{rest}}\frac{E_o}{E_s} \leq 1 \tag{9}$$

and the maximum frequency interval over which locking will occur is:

$$|\Delta f|_{max} = \frac{f_{rest}}{Q}\frac{E_s}{E_o} \tag{10}$$

Assuming that the phase shift α is kept under ±60° the injection lock time constant τ_i approaches

$$\tau_i \sim \frac{E_o Q}{\pi E_s f_{rest}} \qquad \text{(See Eq. 8)} \tag{11}$$

and the condition for synchronization (9) reduces to

$$\tau_i \leq \frac{1}{2\Delta f\pi} \tag{12}$$

Note that only in the formulation of the loop time constant was an angle variation of less than $\pm(n+1)\times 60$ degrees assumed. Linearization is based upon the distribution of a possible 360 degree phase variation between a conventional phase detector and one or more voltage controlled oscillator phase shifters. Furthermore, separation of the injection lock transient from the operation of the phase detector loop assumed that the injection lock process was much faster than the response of the loop (i.e. $\tau_i \ll \tau_L$). While this assumption is useful in the analysis and understanding of the operation of the extended range phase detector it is certainly not a necessary condition for successful operation.

THE LOCK LOCK LOOP

The extended range phase detector may be substituted for a conventional phase detector with little increase in circuit complexity. One rather useful result of such a substitution is in the phase locked loop. There have been a great variety of attempts (e.g. references 7,8) to reduce the threshold signal to noise ratio in FM feedback demodulators and tracking filters by extending the range of the phase detector. However, almost all have destroyed the basic simplicity and elegance of the phase lock technique in the process. Although separate oscillators may be utilized (in both the extended range phase detector and conventional phase locked loop), it is possible to share a single oscillator. Fortunately, the sinusoidal nonlinearity which occurs in a conventional phase detector may be cancelled by the nonlinearity inherent in the operation of an injection locked oscillator phase shifter. The resultant combination, i.e. the Lock Lock Loop (LLL) is a modified phase locked loop. It differs from a conventional phase locked loop in that the frequency and phase of the voltage controlled oscillator is effected not only by the control loop, but also by direct synchronization with the input signal.

Figure 2 is a block diagram of the Lock Lock Loop demodulator. Operation of the loop is best described by first considering the output of the phase detector under the open loop condition illustrated by Figure 3. This open loop operation demonstrates the linear relationship which results from dual synchronization. In this circuit, the injected synchronizing signal pulls the oscillator to the input signal frequency within a time which is short compared to the characteristic modulation, but long compared to one carrier period.

As described previously, the sine of the phase difference between the input signal and the synchronized oscillator is proportional to the difference between the input and oscillator rest frequencies.

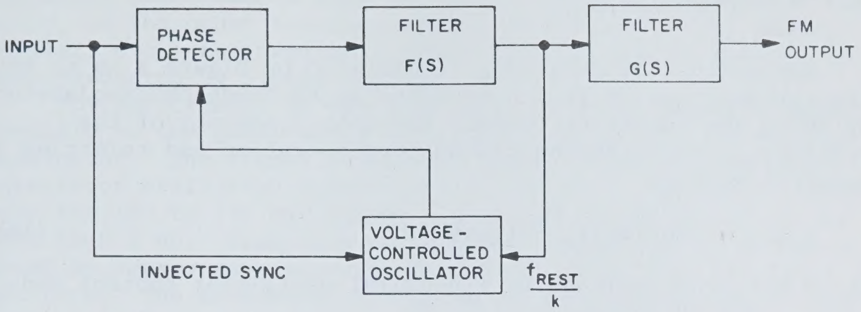
$$\sin(\theta_i - \theta_{osc}) = K(f_i - f_{rest})$$

where

$K = 2QE_o/E_s f_{rest}$

θ_i, θ_{osc} are the phase angles of the input and locked oscillator output

The output of the conventional phase comparator is proportional to the sine of the phase difference between two input signals. Therefore, the output from the phase detector is (from equation 13) proportional to the difference between input and oscillator rest frequencies. This is the key to the improvement offered by the Lock Lock Loop. Two nonlinear operations compensate for one another.


INPUT
PHASE DETECTOR
FILTER
F(S)
FILTER
G(S)
FM OUTPUT
VOLTAGE-CONTROLLED OSCILLATOR
INJECTED SYNC
$\frac{f_{REST}}{k}$

$$\text{FM Output} = G(s)\, f_{rest}$$

$$f_{rest} = k\, K_o F(s)\, (f_i - f_{rest})$$

$$\therefore \text{FM Output} = \frac{K_o G(s)\, F(s)}{1 + k\, K_o F(s)}\, f_i$$

FIGURE 2. THE LOCK LOCK LOOP

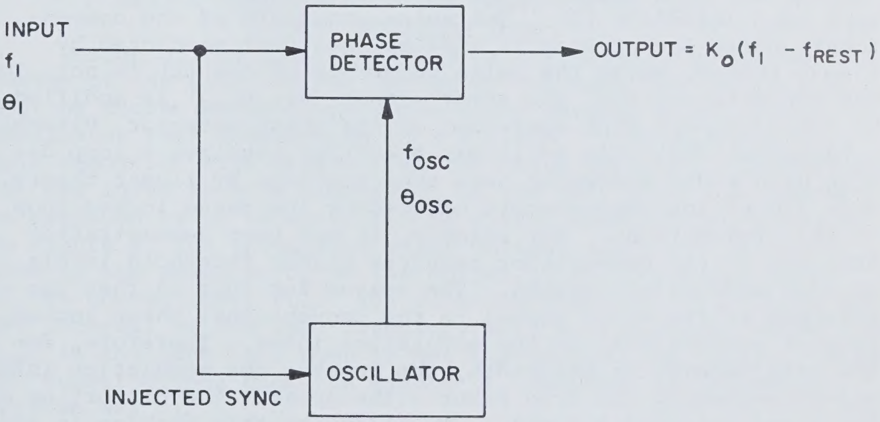

INPUT
f_I
θ_I
PHASE DETECTOR
OUTPUT = $K_o(f_I - f_{REST})$
f_{OSC}
θ_{OSC}
OSCILLATOR
INJECTED SYNC

FIGURE 3. OPEN LOOP DISCRIMINATOR

$$\text{FM Output} = K_o(f_i - F_{rest}) \tag{13}$$

where

$$K_o = \gamma E_o E_S K = \frac{2Q\gamma E_o^2}{f_{rest}}$$

Basically, the open loop illustrated by Figure 3 is an FM discriminator which is converted to an FM-feedback demodulator, by using the output to control the rest frequency of the oscillator. Considering closed loop operation and referring to Figure 2 we see

$$f_{rest} = kK_oF(s)(f_i - f_{rest}) \tag{14}$$

where k is the (voltage to frequency) oscillator control constant. The FM output is given by

$$\text{FM OUTPUT} = \frac{G(s)\,f_{rest}}{k} \tag{15}$$

Combining equation 15 with the solution to equation 14 results in the relation between the input and FM demodulated output.

$$\text{FM OUTPUT} = \frac{K_oG(s)\ F(s)}{1 + kK_oF(s)}\ f_i \tag{16}$$

The loop filter, F(s), is chosen to minimize the tracking error from a noisy input signal. The output filter, G(s), is chosen to optimally demodulate the input signal. The same is true of the conventional phase locked loop "linear model", but note that no first order linearizing approximations have been made in the derivation of the describing equation of the Lock Lock Loop (equation 16). The noise bandwidth of the conventional phase locked loop is greater than that predicted by linear theory, while the noise bandwidth of the LLL is not. In the conventional PLL, the error signal ($\theta_i - \theta_{osc}$) is modified by the nonlinear sine operation of the phase detector. Viterbi[4] has shown that this nonlinear operation requires a loop design with wider bandwidth than that required by linear theory. This limits the improvements offered by the phase locked loop in all applications. For example, it has been demonstrated[5] that the FM-PLL demodulator requires higher threshold levels at high modulation indices. The reason for this is that the strength of the error signal in the conventional phase locked loop is proportional to the modulation index. Therefore, for the same modulation bandwidth, the greater the modulation index and subsequently the true error - the greater the distortion of the error correcting signal. To alleviate this problem in conventional phase locked loops, the loop bandwidth is opened beyond the point predicted by linear filter theory. This reduces the transient error, but increases the noise error and limits the threshold improvement below the best (linear) predicted theoretical result. The Lock Lock Loop with its two compensating nonlinearities, operates as a linear device and therefore, is capable of achieving the theoretically predicted threshold improvements even for large modulation indices. Similarly, in the other tracking filter applications (AM, phase, doppler, etc.) the required noise bandwidth may be reduced to that predicted by linear theory.

The performance of the Lock Lock Loop was compared experimentally to the performance of a conventional delay line discriminator. The signal source consisted of a varactor-tuned transistor oscillator operating at a center frequency of 160MHz. Over the 150 to 170 MHz range, the output power variation was less than 1 db. Frequency shift-keyed test signals were produced by square-wave modulation of the signal frequency at a 10 KHz rate. The modulation bandwidth of the signal was 233 KHz. Peak to peak deviations of 3.5 and 20 MHz were used, corresponding to modulation indices (m) of 7.5 and 43 respectively. Receiver noise was produced by cascading two high gain 160 MHz amplifiers. For each case, the noise bandwidth was restricted to the peak to peak deviation by a bandpass filter.

A threshold extension of 9 db was obtained for the signal having a modulation index (m) of 7.5. This compares favorably with the maximum possible extension (10 log (m + 1)) of 9.3 db. Threshold extension for the signal with a modulation index of 43 was about 15 db as compared to the maximum possible extension of 16.4 db. In the second case the improvement might have been greater than 15 db but could not be measured because of the difficulties encountered in taking accurate data at output signal to noise ratios above 30 db.

DESIGN COMPARISON OF THE PHASE-LOCKED LOOP AND LOCK-LOCK LOOP

Design of the Lock Lock Loop is intimately related to the design of the PLL. In the design of the Lock Lock Loop, F(s) must be chosen so that the total error between input and oscillator rest frequencies is a minimum. From equation 1, this condition will ensure that the phase error between oscillator and input is also a minimum. If the Lock Lock Loop filter is replaced by an integrator-loop filter combination (as shown in Figure 4) the integrated error voltage (point 4) is analogous to the phase error at the output of a "linearized model" of the conventional PLL. The error between the input phase and the integral of the rest frequency must be a minimum. (note that θ_{osc} in the Lock Lock Loop is not the integral of f_{rest}.) With the substitutions shown in Figure 4, equation 16 may be rewritten as:

$$\text{FM OUTPUT} = \frac{K_o G_{PLL}(s)\, F_{PLL}(s)}{s + kK_o\, F_{PLL}(s)} \qquad (17)$$

This is precisely the form of the transfer function of a linearized model of a Phase Locked Loop. Although the values of K_o

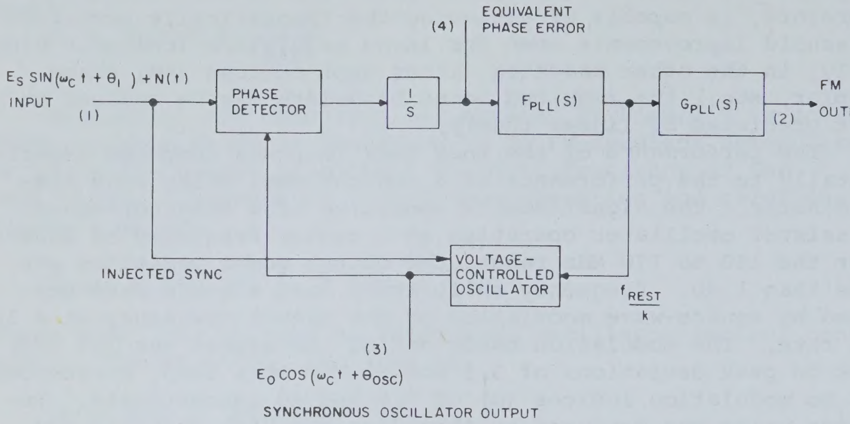

(4)
EQUIVALENT
PHASE ERROR
$E_S \sin(\omega_C t + \theta_I) + N(t)$
INPUT
(1)
PHASE DETECTOR
$\frac{1}{S}$
$F_{PLL}(S)$
$G_{PLL}(S)$
FM
OUTPUT
(2)
INJECTED SYNC
VOLTAGE-CONTROLLED OSCILLATOR
$\frac{f_{REST}}{k}$
(3)
$E_O \cos(\omega_C t + \theta_{OSC})$
SYNCHRONOUS OSCILLATOR OUTPUT

FIGURE 4. COMPARISON OF LOCK LOCK LOOP AND CONVENTIONAL PHASE-LOCKED LOOP

in the linearized Phase Locked Loop and the Lock Lock Loop depend upon different physical parameters, the overall transfer functions of both are identical for the same values of K_O and k. That is, both loops require filters of identical form. The difference between the two loops are listed below:

1. The design for the Lock Lock Loop is exact, the design for the Phase Locked Loop is often based on a linear model.
2. The Lock Lock Loop constant (K_O) is not a function of the signal strength. One need only ensure that the pulling range is of sufficient width to encompass the maximum frequency error.
3. The synchronous output of the Lock Lock Loop (point 3 of Figure 4) is not linearly related to the input unless the pulling range is large compared to the maximum frequency error (that is, $K(f_i - f_{rest}) << 1$).

To summarize these results we may state that the Lock Lock Loop is equivalent to the linear model of the Phase Locked Loop at points 1, 2, and 4, and is approximately equivalent at point 3 if the maximum error between input and rest frequencies is small compared to the pulling range. This is always possible.

CONCLUSIONS

The nonlinear phase shift properties of injection locked oscillators were shown to offer conceptually new hardware solutions to a number of classical problems in the theory of signal processing. However, discussion was limited to those applications which combine the advantages of both phase and injection lock techniques; such as linear and unambiguous phase comparison over the full 360 degree range, FM threshold extension, and tracking filter design. Many other equally new and useful applications are possible - for example, in the design of signal processing antenna arrays such as the phased array and the interferometer.

REFERENCES

1. R. Adler, "A Study of Locking Phenomena in Oscillators," Proc IRE, Vol 34, June 1946.
2. L.J. Paciorek, "Injection Locking of Oscillators," Proc of the IEEE, Vol 53, p 1723-1727, November 1965.
3. M. Sandler, "An Extended Range Phase Detector, " IEEE Symposium Digest, 1970, Canadian Symposium on Communications p 14-15, November 1970.
4. A.V. Viterbi, "Acquisition and Tracking Behaviour of Phase-Locked Loops," Proc of the Symposium on Active Networks and Feedback Systems, Polytechnic Institute of Brooklyn, April 1960.
5. D.L. Schilling and V. Billig, "A Comparison of the Threshold Performance of the Frequency Demodulator Using Feedback and the Phase Locked Loop," Record of the 1965 International Space Electronics Symposium, p 3E1-3E9, 2-4 November.
6. M. Sandler and F. Burnham, "The Lock Lock Loop," NTC '71 Conference Record, p 228-232, April 1971.
7. L.M. Robinson, Tanlock: A Phase Lock Loop of Extended Tracking Capability, Conference Proceeding of the 1962 National Winter Convention on Military Electronics, p 396-407.
8. Acompora & Newton - RCA Review, Vol 27, December 1966, p 577, Use of a phase subtraction to extend the range of a phase locked demodulation.
9. C.L. Beers, "A Frequency-Dividing Locked in Oscillator Frequency-Modulation Receiver," Proc IRE, Vol 32, p 730-738, December 1944.
10. J.R. Woodyard, "Application of the Auto-Synchronized Oscillator to Frequency Demodulation," Proc IRE, p 610-619, May 1937.
11. R.M. Jaffe and E. Rechtin, "Design and Performance of Phase-Lock Circuits Capable of Near Optimum Performance Over a Wide Range of Input Signal and Noise Levels," IRE Trans on Inf Theory, Vol IT-1, No.1, p66-76 March 1955.
12 Hemimam, Newton & Frankle, RCA Review, Vol 29, June 1968, p 252, Multiple Loop Frequency Compressive Feedback for Angle Modulation Detection.

TRANSIENT CHARACTERISTICS OF A HYBRID PHASE-LOCKED-LOOP

T. L. STEWART
Dept. of Electrical Engineering
Bradley University
Peoria, Illinois

W. M. HAMMOND
Dept. of Electrical Engineering
Bradley University
Peoria, Illinois

Summary

A phase-locked-loop is a widely used system to establish phase coherences for coherent demodulation of digital signals. The function of the loop is to provide an estimate of the carrier phase for a signal of the form

$$r(t) = \sqrt{2}\,A \sin(\omega t + \cos^{-1}(m\,x(t)) + \theta(t))$$

$$= \underbrace{\sqrt{2}\,A\,m \sin(\omega t + \theta(t))}_{\text{carrier component}} + \underbrace{\sqrt{2}\,A(1-m^2)^{1/2}\,x(t)\cos(\omega t + \theta(t))}_{\text{data component}}$$

where $x(t)$ is a bi-phase modulated data-subcarrier, m is the modulation index, and $\theta(t)$ represents the variation in the carrier phase and frequency with respect to the receiver. The phase-lock-loop extracts the phase information from the carrier component. It is possible to estimate the carrier phase directly from the data component by using a Costas-type loop.

The hybrid loop is a scheme to use the phase information in both signal components to derive the phase estimate. An obvious advantage is that the total signal power is available which gives improved performance in noise. The steady-state performance of a hybrid-loop in the presence of noise has been analyzed by Stewart(1,2) and Lindsey.(3)

In this paper the transient or acquisition characteristics of the loop are considered. The signal-noise ratio is assumed to be sufficiently large so that noise effects can be neglected. The operating characteristic of interest is the time required to achieve phase-lock from an initial frequency error. The hybrid-loop offers an interesting, additional degree of freedom over a conventional phase-lock-loop or Costas-type loop. These latter schemes give a receiver S-curve that is sinusoidal; however, with the hybrid loop it is possible to alter the shape of the receiver S-curve.

It is well known that the shape of the S-curve can give dramatic improvements in acquisition performance; the Tanlock and Linlock (4) loops are systems which illustrate this point. It is shown that the hybrid-loop can be designed to give similar characteristics and can even be designed to exceed both the Linlock and Tanlock systems with respect to minimizing acquisition time. This is accomplished with a relatively simple adjustment of the loop parameters.

(1) T. L. Stewart, "A Model Distribution for a Hybrid Phase-Locked-Loop," Proceedings of the Third Asilomar Conference on Circuits and Systems December 1969.

(2) T. L. Stewart, "Analysis of a Hybrid Phase-Locked Loop," NASA Technical Note NASA TN D-5666, National Aeronautics and Space Administration, Washington, D.C., June 1970.

(3) L. C. Lindsey, "Hybrid Carrier and Modulation Tracking Loops," IEEE Transactions on Communications Technology, Vol. COM-20 February 1972.

(4) F. G. Splitt, "Design and Analysis of a Linear Phase-Locked Loop of Wide Dynamic Range," IEEE Transactions on Communication Technology, Vol. COM-14, August 1966.

AUTOMATIC EQUALIZATION FOR QUADRATURE AMPLITUDE MODULATION SYSTEMS

Yoichi Sato
Central Research Laboratories
Nippon Electric Co., Ltd.
1753, Shimonumabe, Nakahara-ku, Kawasaki, Japan

Abstract

The zero-forcing equalization method applicable to quadrature amplitude modulation system is derived. A sufficient condition is given in simple form under which the peak distortion is minimized, and it is satisfied for a wide class of distorted quadrature channels. From the above result, a simple equalization algorithm by means of fixed-increment tap adjustment is obtained. The performances of the equalizer are evaluated by computer simulation.

Introduction

As the automatic equalization for baseband pulse amplitude modulation (PAM) systems, the zero-forcing method has been proposed by R.W. Lucky(1). This method is well known for simplicity of instrumentation and minimizing the maximum quantity of intersymbol interference which occurs in the worst case of data sequences. Such a worst case consideration is frequently taken in practical design of a data transmission system.

The quadrature amplitude modulation (QAM) system(2) transmits two independent baseband PAM signals in envelopes of inphase and quadrature carriers, respectively. At the receiver of this type of modulation system, two envelopes are split again by synchronous detector and it is desired for each of the above envelopes not to possess two kinds of intersymbol interference, one caused by the envelope's own data sequence and the other caused by leaks between two data sequences. Thus, both envelopes cannot be equalized separately but can be equalized jointly by two dimensional equalizer.

In this paper, a zero-forcing method extended to such a two dimensional modulation system is developed.

Zero-forcing method concept

In this section, Lucky's zero-forcing method for PAM systems is briefly introduced. The function of automatic equalizer is to reduce intersymbol interference so that data sequence $\{a_k\}$ is detected correctly from received sequence $\{\Sigma a_{k-i} x_i\}$. The transversal equalizer is used in order to transform the sampled sequence of distorted pulse $\{x_k\}$ into a desirable sequence $\{h_k\}$. Then, the maximum of residual intersymbol interference at the equalizer output is given by the peak distortion $\Sigma' |h_k| / |h_0|$ ($=D_h$), where the prime on summations denotes deletion of the k=0 term. The essential points of Lucky's technique minimizing this peak distortion are as follows;

1. If initial peak distortion D_0 $(=\Sigma'|x_k|/|x_0|)$ is less than 1, minimum of D_h occurs when $h_k=0$ $(k\in K_n, k\neq 0)$, where K_n is a set of n successive integers containing zero and each K_n entry corresponds to the location of tap gains.

2. If $D_0<1$, fixed increment, $-\Delta\text{sgn}(h_d)$ of the dth tap gain, reduced D_h.

It should be noted that sufficient condition, $D_0<1$, is broad enough against weakly distorted channels, but, otherwise, the channel must be preequalized before applying the zero-forcing equalizer. Application of the above results for QAM systems are developed in succeeding sections.

Baseband equalization for QAM systems

Let $\{a_k\}$ and $\{b_k\}$ be independent data sequences and $\{x_k\}$ and $\{y_k\}$ be output sequences of synchronous detector shown in Fig.1. In QAM systems, the functional form of $x_k + jy_k$ can be written as(3)

$$x_k + jy_k = \Sigma(a_{k-i} + j\,b_{k-i})X_i \qquad (1)$$

$$X_k = [\Sigma m_{k-i}(p_i + j\,q_i)]\exp(j\theta) \qquad (2)$$

where $\{m_k\}$ is a sequence of sampled values of baseband pulse shaped in the sending terminal, $\{p_k+jq_k\}$ is that of impulse response of equivalent baseband channel of FDM transmission media and θ is constant carrier phase. Under the particular situation where the carrier phase θ is controlled to be zero and frequency domain characteristics of the equivalent baseband channel are completely symmetric in loss and delay, since $q_k\equiv 0$, two Lucky's zero-forcing equalizers can be properly applied separately for detected envelopes $\{x_k\}$ and $\{y_k\}$. However, in practice $q_k\neq 0$, so it is necessary for an equalizer to transform complex valued sequence $\{X_k\}$ as close as possible to the Nyquist sequence $\{N_k\}$, such as,

$$N_k = 1 + j0\ (k=0),\ 0 + j0\ (k\neq 0) \qquad (3)$$

A transversal equalizer meeting this requirement must have complex valued tap gains. The function of this equalizer is written as

$$H_k = \sum_{i\in K_n} X_{k-i}C_i \qquad (4)$$

where C_i's are complex valued tap gains.

Definition of peak distortion

Let the following formula be chosen as a measure of the maximum of residual intersymbol interference

$$D_H = \Sigma'|H_k|/|H_0| \qquad (5)$$

and let it call peak distortion. This measure puts an upper boundary on the maxima of realizable intersymbol interference in any level assignment scheme of a_k and b_k. To clarify this description, the following example is given. Presume a level assignment $a_k + jb_k = 1+j,\ -1+j,\ 1-j,\ -1-j$. In conventional QAM systems, the detections of a_k and b_k will be performed separately for inphase and quadrature envelopes. In such a detection scheme, the exact evaluation of the maximum of intersymbol interference must be accomplished individually for envelopes. It is given by

$$(\Sigma'|\mathrm{Re}[H_k]| + \Sigma|I_m[H_k]|) \;/\; |\mathrm{Re}[H_0]| \tag{6}$$

To determine whether formula (5) places the upper boundary upon formula (6), assume that $\mathrm{Im}[H_0]$ is forced to be zero. This assumption is practically reasonable, if control of the carrier phase acts correctly. It can be said that the evaluation based on D_H, which leads to $\sqrt{2}\,D_H$ by the formula $\max[a_k+jb_k]D_H$, is greater than the value of formula (6). This relation is always satisfied for any level assignment scheme.

Minimization of peak distortion D_H

There are basically two methods of minimizing D_H. The first is where minimization is performed under the condition that center tap gain C_0 is fixed at $1+j0$ and H_0 is released. The second shows the constraint that $H_0=1+j0$. The difference of properties between the two is sketched in Fig.2. In the first type, sequence $\{H_k\}$ is left free in angular co-ordinate. Therefore, an automatic equalizer requires some additional phase control mechanism. On the other hand, in the second type, $\{H_k\}$ satisfies $H_0=1+j0$, hence the intersymbol interference caused by carrier phase deviation is completely cancelled. This difference becomes very significant in demodulator construction.

To solve the above two types of minimization problems, two zero-forcing method are derived. At first, consider the general error estimation function,

$$J = \sum_{k\in K_n} \Big|\sum_{i\in K_n} X_{k-i}C_i - R_k\Big| = \sum_{k\in K_n} |E_k| \tag{7}$$

where $\{R_k\}$ is a reference sequence chosen arbitrarily. The zero-forcing method for minimizing J is obtained in the following theorem.

[Theorem] Let SC'(2) denote the following condition,

$$D_X < (|X_0^2-X_1X_{-1}|+|X_1|+|X_1X_{-1}|)/(1+|X_1|) \tag{8}$$

and

$$D_X < (|X_0^2-X_1X_{-1}|+|X_{-1}|+|X_1X_{-1}|) \;/\; (1+|X_{-1}|) \tag{9}$$

where $D_X=\Sigma'|X_k|/|X_0|$ and $|X_0|=1$ is assumed for convenience. Then, under SC'(2), the minimum of J occurs, if, and only if, $H_k=0+j0$ for all $k\in K_n$ (see Appendix A for the proof).

Since SC'(2) has simple form, it can be easily examined for $\{X_k\}$ to determine whether J can always be minimized by the zero-forcing method or not. Moreover, allowable $\{X_k\}$ satisfying SC'(2) covers a fairly wide range. As the right hand side of inequalities (8) and (9) are greater than 1, so SC'(2) includes the class of $\{X_k\}$ allowed by the condition $D_X<1$. In Table 1, for the special case where $\{X_k\}$ is real valued, examples of upper boundaries on D_X given by SC'(2) are shown, but the cases where X_1 has the same polarity as X_{-1} are omitted, since SC'(2) is equivalent to the condition $D_X<1$. From the theorem, two types of zero-forcing method, which minimize D_H under SC'(2), can be readily derived (see Appendix B).

[First zero-forcing method] Adjust n-1 tap gains, except for the fixed center tap gain C_0, so that

$$H_k = 0 + j0 \quad (k \in K_n,\ k \neq 0) \tag{10}$$

[Second zero-forcing method] Adjust all n tap gains, so that

$$\left.\begin{aligned} H_k &= 0 + j0 \quad (k \in K_n,\ k \neq 0) \\ H_0 &= 1 + j0 \end{aligned}\right\} \tag{11}$$

Fixed-increment tap adjustment algorithm

Several structures of automatic equalizer which minimize D_H under SC'(2) can be designed by use of the well-known numerical calculation techniques of solving a set of simultaneous linear equations, such as Eq.(10) or (11). The algorithm will be chosen properly against the practical requirements of rapidly setting up, simple implementation or weak condition of convergence. Especially when simple implementation is desired, the algorithm of tap gain adjustment by unit quantity will be efficient. An example of such algorithms is proposed here. Figure 3 shows the present rule. By this rule, the following error estimation function for the general case with arbitrary reference sequence,

$$J' = \sum_{k \in K_n} \left| \sum_{i \in K_n} X_{k-i} C_i - R_k \right| \tag{12}$$

can be reduced under sufficient conditions,

$$D_x < \sqrt{2}^{-1} \tag{13}$$

and

$$|E_d|/\Delta > \sqrt{2}^{-1}\ (1-D_x^2)/(1-\sqrt{2}\ D_x) \tag{14}$$

(see appendix C). It should be noted that the above additional conditions narrow the range convered by SC'(2). To see how wide the range of condition (13) is, consider the following minimax formula for the received envelope $\{x_k\}$ (or $\{y_k\}$)

$$\min_{\theta}\ [(\Sigma'|\mathrm{Re}[X_k]|+\Sigma|\mathrm{Im}[X_k]|)/|\mathrm{Re}[X_0]|] \tag{15}$$

This minimum occurs when carrier phase θ is controlled as $\mathrm{Im}[X_0]=0$, and it becomes a quantity which cannot be removed without an equalizer. In this sense, formula (15) can be regarded as the initial peak distortion evaluated separately in envelopes. Now it is shown that formula (15) is less than $\sqrt{2}\ D_x$. Thus, if $D_x<\sqrt{2}^{-1}$, formula (15) is less than 1. Therefore, against the distortion remaining after carrier phase is properly controlled, it can be said that the condition $D_x<\sqrt{2}^{-1}$ is equivalent to Lucky's condition $D_0<1$, applied to $\{x_k\}$ (or $\{y_k\}$). Second condition (14) implies that the error $|E_d|$ $(d \in K_n)$ after convergence does not exceed $(\Delta/\sqrt{2})(1-D_x{}^2)/(1-\sqrt{2}D_x)$. Figure 4 shows the region of $|E_d|/\Delta$ and D_x satisfying condition (14).

Numerical evaluation

Computer simulation is performed by use of the frequency domain characteristics taken from an actual telephone line having 3 links. The modulation rate is fixed at 1.6 k baud and the carrier frequency at 1.8 kHz. Baseband pulse m(t) is shaped in 50% cosine roll off. Then equivalent baseband impulse response X(t) is as shown in Fig.5(a). Let the center of X(t) be determined at the time of max|X(t)|. Then the initial peak distortion D_X is 0.531, and, after zero-forcing, D_H is reduced to 0.057 for 7 taps, 0.033 for 9 taps,, 0.019 for 11 taps, 0.011 for 13 taps and 0.007 for 15 taps. Figure 5(b) shows the equalized pulse H(t) for 7 taps. Figure 6 illustrates the convergence processes of the fixed increment algorithm in the previous section, where the initial state has been chosen as $C_0=X_0^*$, $C_k=0+j0$ ($k\in K_n$, $k\neq 0$) and the number of taps has been 7.

Conclusion

The zero-forcing equalization methods for QAM systems are proposed. The main results are as follows.

(1) Two types of zero-forcing method have been derived. They will have significant different features in demodulator construction. (2) A sufficient condition for the zero-forcing method has been derived and its applicable region has been investigated. (3) the theorem is applicable to the general case of arbitrary reference sequence. (4) Fixed-increment algorithm has been proposed for instrumentation simplicity. (5) The performance of the present zero-forcing method have been evaluated by computer simulation.

The future problem will be the application of this zero-forcing method to the adaptive type of equalizer.

Acknowledgement

The author was favored by the assistance of Mr. Kojiro Watanabe, who contributed his simulation and helpful discussion, and wishes to thank Mr. Tatsuo Ishiguro, Dr. Yoshihiro Iwadare and Dr. Hisashi Kaneko for their helpful advice.

References

1. R.W. Lucky, "Automatic Equalization for Digital Communication" BSTJ, vol.44, April 1965, pp 547-588.
2. R.W. Lucky, J. Salz and E.J. Weldon,Jr., "Principles of Data Communication" New York, McGraw-Hill, 1968.
3. Hisashi Kobayashi, "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems", IEEE trans., vol. COM-19, No.3, June 1971, pp 268-293.

Appendix A

Let entries of K_n be(-i, ..., 0, ..., n-i-1)and define the following n dimensional vectors and matrix,

$$\underline{x}_n^k = \begin{pmatrix} X_{k+i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ X_{k-n+i+1} \end{pmatrix} \quad \underline{r}_n = \begin{pmatrix} R_{-i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ R_{n-i-1} \end{pmatrix} \quad \underline{e}_n = \begin{pmatrix} E_{-i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E_{n-i-1} \end{pmatrix} \quad \underline{c}_n = \begin{pmatrix} C_{-i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ C_{n-i-1} \end{pmatrix}$$

$$\underline{v}_n^j = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \left.\vphantom{\begin{matrix}0\\ \cdot\\ \cdot\\ 0\end{matrix}}\right\} i+j \qquad \underline{X}_n = \begin{bmatrix} X_0 & X_{-1} & & X_{-n+1} \\ X_1 & X_0 & & X_{-n+2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ X_{n-1} & X_{n-2} & \cdots\cdots & X_0 \end{bmatrix}$$

Preceeding proof for the theorem, two lemmas are shown.

[Lemma 1] For all $j \in K_n$, let the inequality

$$0 < 1 - \sum_{k \notin K_n} |(\underline{x}_n^k, \underline{X}_n^{-1} \underline{v}_n^j)| \qquad \text{(A1)}$$

be satisfied, where (,) denotes scalar product, then minimum of J occurs if, and only if, $E_k = 0 + j0$ for all $k \in K_n$.

(proof) Splitting J into two terms of $k \notin K_n$ and $k \in K_n$, then

$$J = \sum_{k \notin K_n} |E_k| + \sum_{k \in K_n} |E_k| \qquad \text{(A2)}$$

Since E_k ($k \notin K_n$) can be written in terms of the entries of $\underline{e}_n$ as $E_k = (\underline{x}_n^k, \underline{X}_n^{-1} \underline{e}_n) + (\underline{x}_n^k, \underline{X}_n^{-1} \underline{r}_n) - R_k$, substituting this into the first term in the right hand side of Eq. (A2), we obtain

$$J = \sum_{k \notin K_n} |(\underline{x}_n^k, \underline{X}_n^{-1} \underline{e}_n) + (\underline{x}_n^k, \underline{X}_n^{-1} \underline{r}_n) - R_k| + \sum_{k \in K_n} |E_k| \qquad \text{(A3)}$$

Extracting E_k ($k \in K_n$) repeatedly out of the first term in the above right hand side,

$$J = \sum_{k \notin K_n} \xi_k |(\underline{x}_n^k, \underline{X}_n^{-1} \underline{r}_n) - R_k| + \sum_{j \in K_n} \left(1 + \sum_{k \notin K_n} \sigma_k^j |(\underline{x}_n^k, \underline{X}_n^{-1} \underline{v}_n^j)|\right) |E_j| \qquad \text{(A4)}$$

is obtained. 1 or -1 are assigned to ξ_k and σ_k^j, depending on their subscripts k and j. Now it can be concluded that, if inequality (A1) is satisfied for all $j \in K_n$, every coefficient of $|E_j|$ must be positive and, therefore, if, and only if, $E_k = 0 + j0$ for all $k \in K_n$, J becomes minimum. (Q.E.D.)

[Lemma 2] Denote the sufficient condition given by inequality (A1) in lemma 1 by SC(n). If SC(n) is satisfied for the sequence $\{X_k\}$, SC(n+1) for the set K_{n+1}, including K_n, is also satisfied for the same sequence $\{X_k\}$.

(proof) Let the entries of K_{n+1} be (-i, ..., 0, ..., n-i) and denote $\sum_{k \notin K_n} |(\underline{x}_n^k, \underline{X}_n^{-1} \underline{v}_n^j)|$, which depends on n and j, by G_n^j. Then $G_n^j < 1$ $(j \in K_n) \Rightarrow G_{n+1}^j < 1$ $(j \in K_{n+1})$ proves the lemma.

(1) At first, in cases of $j \in K_n$, let us derive the relation between G_n^j and G_{n+1}^j. Presume the linear equations for n+1 taps,

$$\underline{X}_{n+1} \underline{c}_{n+1}^j = \underline{v}_{n+1}^j \quad (j \in K_n) \qquad \text{(A5)}$$

The solution $\underline{C}_{n+1}^j$ can be written by use of the solution $\underline{c}_n^j = \underline{X}_n^{-1} \underline{v}_n^j$ of n taps, as

$$\underline{c}^j_{n+1} = \Delta\underline{c}^j_{n+1} + (\underline{c}^j_n|0) \qquad \text{(A6)}$$

where $(\underline{c}^j_n|0)$ represents the n+1 dimensional vector generated by concatenating $\underline{c}^j_n$ and zero. Substituting Eq. (A6) into Eq. (A5), $\Delta\underline{c}^j_{n+1}$ is derived as

$$\Delta\underline{c}^j_{n+1} = -\underline{X}^{-1}_{n+1}\underline{v}^{n-i}_{n+1}(\underline{x}^{n-i}_n, \underline{X}^{-1}_n\underline{v}^j_n) \qquad \text{(A7)}$$

On the other hand, the following inequality is obtained for all $j \in K_n$.

$$G^j_{n+1} = \sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, \Delta\underline{c}^j_{n+1} + (\underline{c}^j_n|0))$$

$$\leqslant \sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, \Delta\underline{c}^j_{n+1})| + \sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, (\underline{c}^j_n|0))| \qquad \text{(A8)}$$

Substituting Eq. (A7) and $\underline{c}^j_n=\underline{X}^{-1}_n\underline{v}^j_n$ into Eq. (A8),

$$G^j_{n+1} \leqslant |(\underline{x}^{n-i}_n, \underline{X}^{-1}_n\underline{v}^j_n)| \sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, \underline{X}^{-1}_{n+1}\underline{v}^{n-i}_{n+1})|$$

$$+ \sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, (\underline{X}^{-1}_n\underline{v}^j_n|0))| \qquad \text{(A9)}$$

Furthermore, the second term in the above right hand side

$$\sum_{k\notin K_{n+1}} |(\underline{x}^k_{n+1}, (\underline{X}^{-1}_n\underline{v}^j_n|0)) = \sum_{k\notin K_{n+1}} |(\underline{x}^k_n, \underline{X}^{-1}_n\underline{v}^j_n)|$$

$$= \sum_{k\notin K_n} |(\underline{x}^k_n, \underline{X}^{-1}_n\underline{v}^j_n)| - |(\underline{x}^{n-i}_n, \underline{X}^{-1}_n\underline{v}^j_n)| = G^j_n - |(\underline{x}^{n-i}_n, \underline{X}^{-1}_n\underline{v}^j_n)| \qquad \text{(A10}$$

is derived. Now, denoting $|(\underline{x}^{n-i}_n, \underline{X}^{-1}_n\underline{v}^j_n)|$ by α^j_{n-i} and substituting Eq. (A10) into Eq. (A9), the following desired relation was obtaine

$$G^j_{n+1} \leqslant \alpha^j_{n-i}G^{n-i}_{n+1} + G^j_n - \alpha^j_{n-i} \quad (j\in K_n) \qquad \text{(A11}$$

(2) Second, when j=n-i, namely $j\in K_{n+1}$ and $j\notin K_n$, the relation between G^{n-i}_n and G^{n-i}_{n+1} can be derived through the same process as used in the above derivation. The result is

$$G^{n-i}_{n+1} \leqslant \alpha^{n-i-1}_{-i-1}G^{-i}_{n+1} + G_n^{n-i-1} - \alpha^{n-i-1}_{-i-1} \qquad \text{(A12)}$$

Now the lemma from two relations, (A11) and (A12), can be proved as follows. Substitute j=-i($-i\in K_n$) into relation (A11), then

$$G^{-i}_{n+1} \leqslant \alpha^{-i}_{n-i}G^{n-i}_{n+1} + G^{-i}_n - \alpha^{-i}_{n-i} \qquad \text{(A13)}$$

Both G^{n-i}_{n+1} and G^{-i}_{n+1}, which simultaneously satisfy relations (A12) and (A13), must be less than 1, since G^{n-i-1}_n and G^{-i}_n are less than 1 from the assumption and evidently $0\leqslant\alpha^{n-i-1}_{-i-1}\leqslant G^{n-i-1}_n$, $0\leqslant\alpha^{-i}_{n-i}\leqslant G^{-i}_n$. For another G^j_{n+1}($j=-i+1, \ldots, 0, \ldots, n-i-1$), it can be shown from relation (A11) that they are all less than 1, because of $0\leqslant \alpha^j_{n-i}\leqslant G^j_n$, assumption $G^j_n<1$ and $G^{n-i}_{n+1}\leqslant 1$ obtained just above. (Q.E.D.

(Proof for the theorem) From the two lemmas, we can chose SC(2) instead of SC(n). SC(2) is written as

$$\Sigma|X_kX_0-X_{k-1}X_1| < 2|X_0^2-X_1X_{-1}| \qquad \text{(A14}$$

and

$$\Sigma|X_kX_{-1}-X_{k-1}X_0| < 2|X_0^2-X_1X_{-1}| \qquad \text{(A15}$$

From this, the form by which the upper limit of D_x is given, can be obtained. For convenience, assume $|X_0|=1$, then the left hand side of inequality (A14), satisfies the following,

$$\Sigma|X_kX_0-X_{k-1}X_1| \leqslant (1+|X_1|) \sum_{|k|\geqslant 2} |X_k|+|X_{-1}|+|X_0^2-X_1X_{-1}|+|X_1^2|$$

$$= (1+|X_1|)(D_x-|X_{-1}|-|X_1|)+|X_{-1}|+|X_0^2-X_1X_{-1}|+|X_1^2|$$

$$= (1+|X_1|)D_x-|X_1|-|X_1X_{-1}|+|X_0^2-X_1X_{-1}| \tag{A16}$$

Substituting this into inequality (A14), the desired inequality (8) in the theorem is obtained. Similarly, from inequality (A15), inequality (9) can be obtained. (Q.E.D.)

Appendix B

Consider the case that $R_0=1+j0$, $R_k=0+j0$, $(k\neq 0)$ in the general error estimation function J. It can be said from Eq. (A4) that, under SC'(2) the minimum of J occurs when $E_k=0+j0$ $(k\in K_n,\ k\neq 0)$ and then E_0 is permitted to be fixed anywhere. Let the solution of this case be $\underline{c}^{\dagger}$, then the constant times of $\underline{c}^{\dagger}$ is a general solution which minimizes J under an arbitrarily fixed E_0. On the other hand, D_H does not change, even if all tap gains are changed by constant times. Moreover, it is shown that a solution which minimizes D_H can be found in general solution which minimizes J. From the above descriptions, it can be said that the constant times of $\underline{c}^{\dagger}$ equal the solution which minimizes D_H. Therefore, under SC'(2), minimum of D_H occurs when $E_k=0+j0$ $(k\in K_n,\ k\neq 0)$.

Appendix C

Denote by J_d' the error estimation function J' after C_d changes by δ_d, then

$$J_d' \leqslant \sum_{\substack{k\in K_n\\ k\neq d}} |E_k| + \sum_{\substack{k\in K_n\\ k\neq d}} |X_{k-d}\delta_d| + |E_d+X_0\delta_d|$$

$$\leqslant J' - (|E_d|-D_x|\delta_d|-||E_d|+X_0E_d^*|E_d|^{-1}\delta_d|) \tag{A17}$$

where * represents the complex conjugate mark. According to the rule shown in Fig.3, the maximum of the last term in above parenthesis is $\sqrt{\Delta 2+|E_d|^2-\sqrt{2}\Delta|E_d|}$, where $|X_0|=1$ is assumed. Hence, in order to reduce J', the following inequality must be satisfied.

$$|E_d| - \Delta D_x - \sqrt{\Delta^2+|E_d|^2-\sqrt{2}\Delta|E_d|} > 0 \tag{A18}$$

From inequality (A18), we can obtain inequalities (13) and (14).

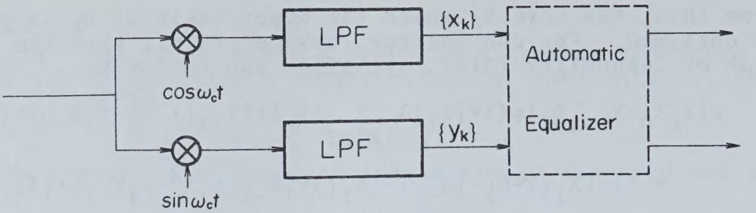

LPF
LPF
$\{x_k\}$
$\{y_k\}$
Automatic
Equalizer
$\cos\omega_c t$
$\sin\omega_c t$

Fig 1. Synchronous detector

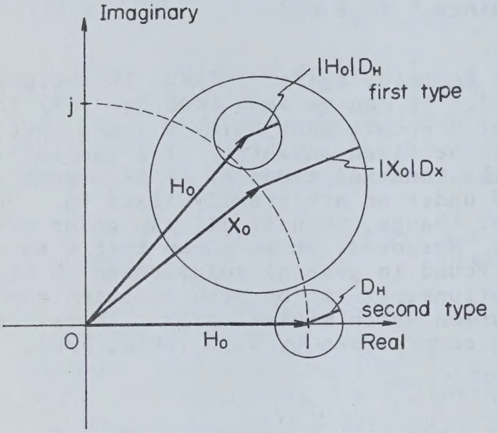
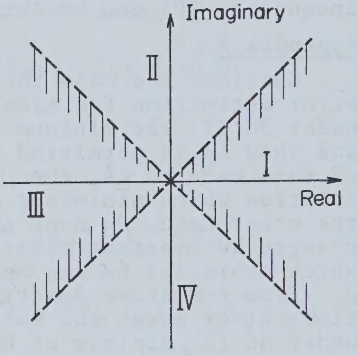

Imaginary
j
$|H_0|D_H$
first type
$|X_0|D_x$
H_0
X_0
D_H
second type
O
H_0
1
Real

Fig 2. Difference between two types of minimizing D_H


Imaginary
II
I
III
Real
IV

$$\delta_d = \begin{cases} \Delta & (X_0^* E_d \in \mathrm{III}) \\ -j\Delta & (\quad " \quad \in \mathrm{IV}) \\ -\Delta & (\quad " \quad \in \mathrm{I}) \\ j\Delta & (\quad " \quad \in \mathrm{II}) \end{cases}$$

Fig 3. Rule for detecting an increment of C_d

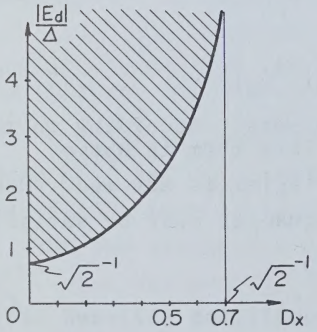


Fig 4. Region of D_x and $\frac{|E_d|}{\Delta}$ satisfying Eq. (14).

X_1 \ X_{-1}	-0.2	-0.4	-0.6	-0.8	-1.0
0.2	1.067	1.114	1.150	1.178	1.200
0.4	1.114	1.229	1.300	1.356	1.400
0.6	1.150	1.300	1.450	1.533	1.600
0.8	1.178	1.356	1.533	1.711	1.800
1.0	1.200	1.400	1.600	1.800	2.00

Table 1. Value of D_X upper bounded by $Sc'(2)$

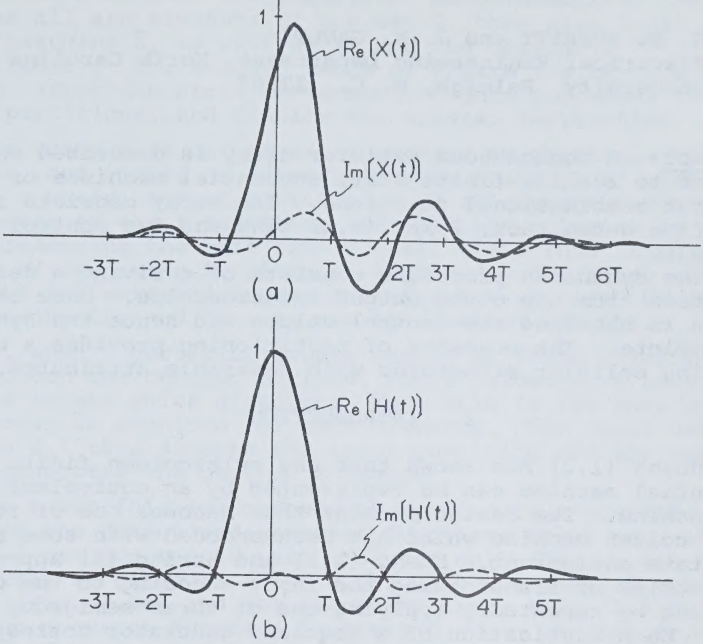


Fig 5. Examples of X(t) and H(t).

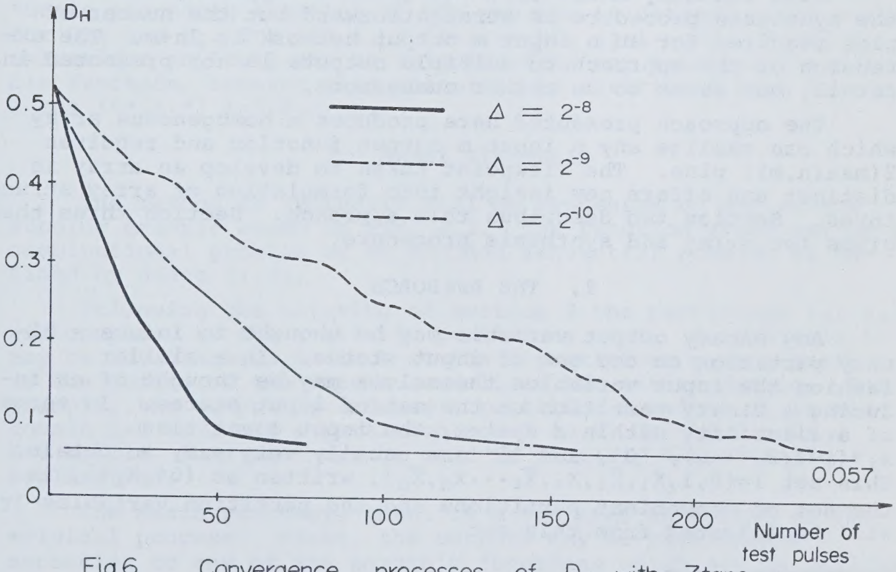


Fig 6. Convergence processes of D_H with 7taps.

A CELLULAR REALIZATION OF FINITE STATE MACHINES USING PARTITIONS

S. F. AVERITT and J. W. GAULT*
Electrical Engineering Department, North Carolina State
University, Raleigh, N. C. 27607

Abstract: A homogeneous cellular array is described which can be used to realize finite state sequential machines or n input m output combinational functions. The array consists of cells with five gates each, 2 max (m,n) pins and two control points per cell.

The synthesis procedure consists of deriving a desirable partition from the given output relationships. Once this partition is obtained the control values and hence the synthesis is complete. The concepts of partitioning provides a model for deriving cellular structures with desirable attributes.

1. INTRODUCTION

Huang [1,2] has shown that any multicolumn finite state sequential machine can be represented by an equivalent one column machine. The design problem then becomes one of realizing a one column machine which has been encoded with some appropriate state assignment. Huang [1,2] and Haring [3] approached the problem of transforming the input encoding to the output encoding by repeatedly applying one of three semigroup generators. Each application of a required generator corresponds to a row in the array. The bound given on the number of rows required grows rapidly and the method for finding the desired sequence of generators is heuristic.

More recently Akers [4], has given an approach in which the synthesis procedure is straightforward but the number of pins required for an n input m output network is 2n+m. The extension of the approach to multiple outputs is not presented in detail, but seems to be rather cumbersome.

The approach presented here produces a homogeneous array which can realize any n input m output function and requires 2(max(n,m)) pins. The viewpoint taken to develop an array is distinct and offers new insight into formulation of array structures. Section two describes this approach. Section three then gives the array and synthesis procedure.

2. THE APPROACH

Any binary output variable may be thought to induce a binary partition on the set of input states. In a similar fashion the input variables themselves may be thought of as inducing a binary partition on the set of input states. In terms of availability within a system, the input quantities $X_i*(i=1,2,\cdots n)$, '0', and '1' are usually very easy to obtain. This set $I=\{0,1,X_1,\overline{X}_1,X_2,\overline{X}_2\cdots X_n,\overline{X}_n\}$, written as $\{0*,X_i*\}$, is the set of convenient partitions and the partition variables (Y) will be selected from this set.

---

*This work was done, in part, while this author was with the Army Research Office, Durham, North Carolina.

If a particular state assignment produces an input to output encoding such that the partitions associated with the output variables all are elements of the set I, then zero logic is required (assuming $\overline{X}_i$ as well as X_i and the constants '1', '0' are all available). The encoded table of Figure 1.a is the desired function after the state assignment, Figure 1.b shows the resulting partitions, and finally the trivial realization.

This example is not given as amazing or particularly realistic, but rather as introductory to the idea of what constitutes a useful output partition. If all output partitions are elements of the I partition set then the solution is obvious. Before presenting the more likely case, where this is not true, the form of the array is given in Figure 2.1. The array shown in Figure 2.1 has external inputs $X_1 \cdots X_n$; $Y_1 \cdots Y_m$ are the internal partition inputs, and $Z_1 \cdots Z_m$ are the desired outputs.

Now, when Y partition inputs cannot be found which identically produce the desired outputs, the Y values must be altered for those inputs which disagree. When this is the case one row in the array is required for each mismatch. The input section of Figure 2.1 then detects the input condition and the output section generates the desired Y change.

This basic thinking can now be applied to a modified structure where the detect and generate functions which were shown in Figure 2.1 are now merged into a single cell (Fig. 2.2a) the resultant cell is given in (Fig. 2.2b) with the logic diagram for the cell (Fig. 2.2c) and the resultant array (Fig. 2.2d). When an n input m output function is desired, then the max (m,n) must be used to determine the array dimensions.

At this point it is important to point out a few other properties and requirements for the resultant array. The boundary conditions are fixed at $Y_{1j}=0$, $A_{in}=1$, and $R_{i1}=1$. The synthesis procedure (section 3) now amounts to determing the values for Q_{ij} and P_{ij} for all i, j $(i=1,\cdots k)$, $(j=1\cdots n)$. P and Q values for any cell may be selected from any internally available function, however this selection is restricted here to the I set $\{0^*, X_i^*\}$ $i=1,2,\cdots n$.

3. ARRAY SYNTHESIS

The method of synthesis will be presented by way of the running example shown in Fig. 3.1. The function shown may be a combinational problem or an encoded sequential problem as defined by Huang [1,2].

Following the thinking of section 2 the partitions π_{z1}, π_{z2} are compared to the available partitions of I. In addition it may be advantageous to look at the Z partitions over a portion of the domain of the inputs as well as the entire domain. For example, the output Z_1 induces a partition equal to X_2 in the domain X=0 to 3 and a partition equal to X_1 in the domain X=8 to 11. This fact can be used to simplify the overall realization.

The basic strategy, then, is a two step extension of the original process. First, the outputs may be partitioned according to any of the possible functions which can be realized by the detection function of equation 3.1.

$$(X_1 \oplus Q_1) \bullet (X_2 \oplus Q_2) \bullet \cdots \bullet (X_n \oplus Q_n) = f_d \qquad (3.1)$$

The Q_i values represent control inputs which may be selected as any element of the I set.

Next the functions within a block of the first level partition (f_d) are realized using one of the possible generate function f_g (eq. 3.2). The particular f_g to be used is determined by comparing the outputs, within the domain of a block, with possible generate functions.

$$f_{d_1} \bullet p_{1_j} \oplus f_{d_2} \bullet p_{2_j} \oplus \cdots \oplus f_{d_\ell} \bullet p_{\ell j} = \oint f_{d_i} \bullet p_{i_j} = f_g \qquad (3.2)$$

where f_{d_i} is the detection function (eq. 3.1) for row i and p_{ij} is the control for row i and output j and may be any element of the I set.

Now proceeding with the example, one possible first level partition on z is

$$z = \{\underbrace{0,1,2,3,12,13,14,15}_{}\!\!\!\!\!\!\!\!\overset{B0}{\overline{}}\ \ \overset{B1}{\overline{4,5,6,7,8,9,10,11}}\}$$

which corresponds to $X_1 \oplus X_2 \ \varepsilon \ f_d$. Within block

$$B_0 : \pi z_1 = \{\overline{0,1,14,15},\ \overline{2,3,12,13}\} = \pi_{X_3 \oplus X_1} \varepsilon f_g \text{ and}$$

$$\pi z_2 = \{\overline{0,3,12,15},\ \overline{1,2,13,14}\} = \pi_{X_4 \oplus X_3} \varepsilon f_g .$$

and within block

$$B_1 : \pi z_1 = \{\overline{4,6,7,8,10},\ \overline{5,9,11}\} = \pi_{X_4} \varepsilon f_g \text{ with a discrepancy at}$$

$$\overline{X}_1 X_2 X_3 X_4 \text{ and}$$

$$\pi z_2 = \{\overline{4,5,8,9},\ \overline{6,7,10,11}\} = \pi_{X_3} \varepsilon f_g .$$

A realization of this particular choice of partitions will require four rows. At this point in time a satisfactory algorithm has not been found for finding the "best" z partitions.

Once the z partitions have been found the P_{ij} and Q_{ij} values may be determined. The Q values control the process of detection.

In this example $X_1 \oplus X_2$ establishes the first level partition. B_0 is $\overline{(X_1 \oplus X_2)}$ or $\overline{X}_2 \oplus X_1$ and $\overline{X}_2 \oplus X_1 = (X_1 \oplus \overline{X}_2)(X_2 \oplus \overline{X}_2)$ which is of the form of equation 3.2 and describes the domain of rows 1 and 2, hence $Q_{11} = Q_{12} = Q_{21} = Q_{22} = \overline{X}_2$. B_1, which is the domain of row 3, is $X_1 \oplus X_2 = (X_1 \oplus X_2)(X_2 \oplus \overline{X}_2)$ and hence $Q_{31} = X_2$, $Q_{32} = \overline{X}_2$. The discrepancy in B_1 is handled in row 4 and is $\overline{X}_1 X_2 X_3 X_4 = (X_1 \oplus 1)(X_2 \oplus 0)(X_3 \oplus 0)(X_4 \oplus 0)$ hence $Q_{41} = 1$, $Q_{42} = Q_{43} = Q_{44} = 0$.

The values of p are associated with generation partitions for each of the domains of detection. These partitions are

$$B_0: \quad \pi z_1 \longrightarrow \pi_{X_3 \oplus X_1} \longrightarrow \begin{matrix} p_{11} = X_3 \\ p_{21} = X_1 \end{matrix}$$

$$\pi z_2 \longrightarrow \pi_{X_4 \oplus X_3} \longrightarrow \begin{matrix} p_{21} = X_4 \\ p_{22} = X_3 \end{matrix}$$

$$B_1: \quad \pi z_1 \longrightarrow \pi_{X_4} \longrightarrow p_{31}=X_4$$

$$z_2 \longrightarrow \pi_{X_3} \longrightarrow p_{32}=X_3$$

and the generation of a correction for the discrepancy in B_1 (accomplished in row 4).

$p_{41} = 1$

$p_{42} = 0$

As a short aside, since $\overline{X}_i$ values are assumed to be available, they may be generated one complement per row by selecting for $\overline{X}_i$ $Q_i=1$, $Q_j=\overline{X}_j$ for $j < i$.

4. SUMMARY

The array described above will realize any n input m output function and can be characterized as

i) homogeneous
ii) requiring 2(max (m,n)) pins
iii) having a cell complexity of five gates
iv) requiring no more than $\max(n,m)2^{n-1}$ cells.

The array synthesis is based on the concept of selecting partitions which require a minimum amount of input detection and output generation. Work is continuing on the development of an improved array using these concepts.

5. REFERENCES

1. Huang, J-C., "Cellular Array Realizations of a Finite State Machine", Ph.D. Dissertation, University of Pennsylvania, 1969.

2. Huang, J-C., "A Universal Cellular Array", IEEE Trans. on Computers, March 1971.

3. Haring, D. R., "Sequential Circuit Synthesis State Assignment Aspects", M.I.T. Press, 1966.

4. Akers, S. B., "A Rectangular Logic Array", IEEE Trans. on Computers, August 1972.

	Inputs			Outputs			
	X_1	X_2	X_3	Z_1	Z_2	Z_3	Z_4
0	0	0	0	1	0	1	1
1	0	0	1	1	1	1	1
2	0	1	0	0	0	1	1
3	0	1	1	0	1	1	1
4	1	0	0	1	0	1	0
5	1	0	1	1	1	1	0
6	1	1	0	0	0	1	0
7	1	1	1	0	1	1	0

Function Description
(a)

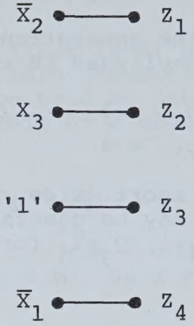
$$\Pi_{Z_1} = \{\overline{2367};\overline{0145}\} = \Pi_{\bar{X}_2}$$

$$\Pi_{Z_2} = \{\overline{0246};\overline{1357}\} = \Pi_{X_3}$$

$$\Pi_{Z_3} = \{\emptyset,\overline{01234567}\} = \Pi_1$$

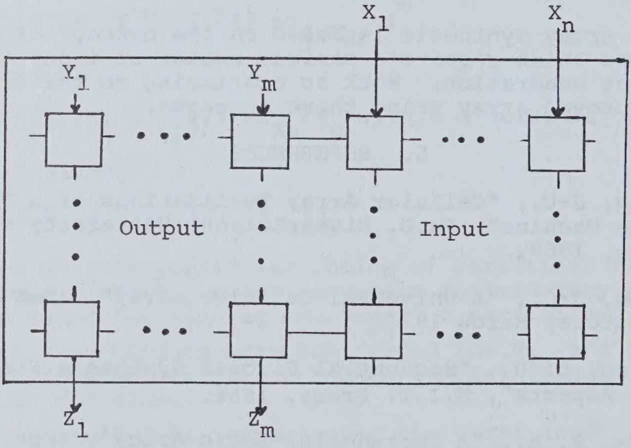
$$\Pi_{Z_4} = \{\overline{4567};\overline{0123}\} = \Pi_{\bar{X}_1}$$

Partitions
(b)


$\bar{X}_2$
Z_1
X_3
Z_2
'1'
Z_3
$\bar{X}_1$
Z_4

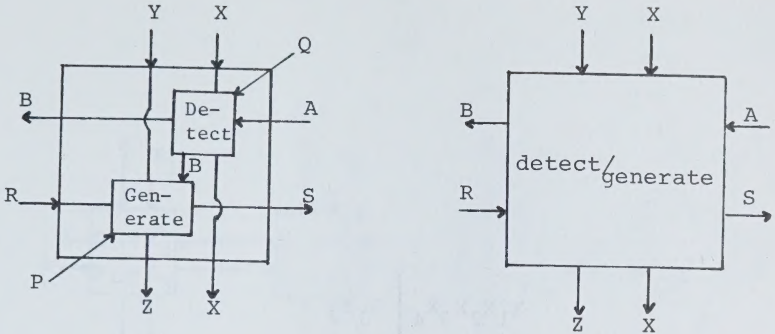
Realization
(c)

Figure 1. Zero Logic Example


X_1
X_n
Y_1
Y_m
Output
Input
Z_1
Z_m

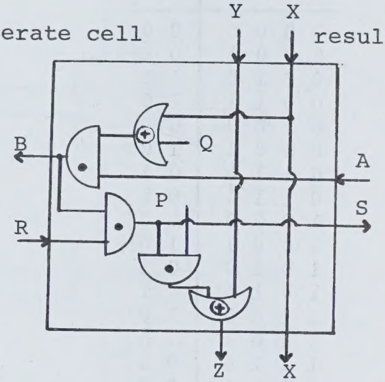
n input m output array

Figure 2.1 The Conceptual Array

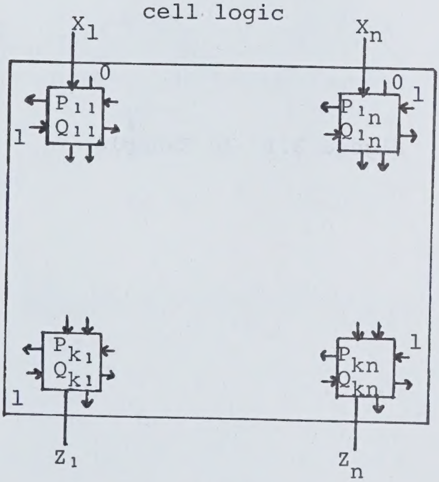

Y
X
Q
B
De-
tect
A
B
R
Gen-
erate
S
P
Z
X
Y
X
B
A
detect/generate
R
S
Z
X

(a)
merged detect/generate cell

(b)
resultant cell


Y
X
B
Q
A
P
S
R
Z
X

(c)
cell logic


X_1
X_n
0
0
1
P_{11}
Q_{11}
1
P_{1n}
Q_{1n}
P_{k1}
Q_{k1}
1
1
P_{kn}
Q_{kn}
Z_1
Z_n

(d)
the array

Figure 2.2 Composite Array

$x_1x_2x_3x_4$	z_1z_2
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 1
0 0 1 1	1 0
0 1 0 0	0 0
0 1 0 1	1 0
0 1 1 0	0 1
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	1 0
1 0 1 0	0 1
1 0 1 1	1 1
1 1 0 0	1 0
1 1 0 1	1 1
1 1 1 0	0 1
1 1 1 1	0 0

Figure 3.1 An Example

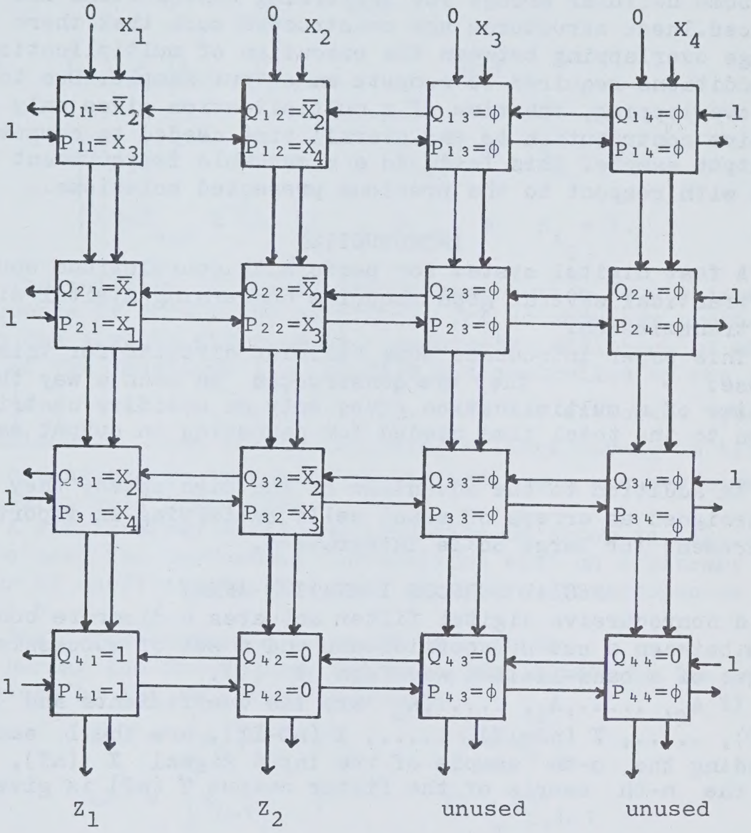

0
X_1
0
X_2
0
X_3
0
X_4
$Q_{11}=\overline{X}_2$
$P_{11}=X_3$
$Q_{12}=\overline{X}_2$
$P_{12}=X_4$
$Q_{13}=\phi$
$P_{13}=\phi$
$Q_{14}=\phi$
$P_{14}=\phi$
$Q_{21}=\overline{X}_2$
$P_{21}=X_1$
$Q_{22}=\overline{X}_2$
$P_{22}=X_3$
$Q_{23}=\phi$
$P_{23}=\phi$
$Q_{24}=\phi$
$P_{24}=\phi$
$Q_{31}=X_2$
$P_{31}=X_4$
$Q_{32}=\overline{X}_2$
$P_{32}=X_3$
$Q_{33}=\phi$
$P_{33}=\phi$
$Q_{34}=\phi$
$P_{34}=\phi$
$Q_{41}=1$
$P_{41}=1$
$Q_{42}=0$
$P_{42}=0$
$Q_{43}=0$
$P_{43}=\phi$
$Q_{44}=0$
$P_{44}=\phi$
1
1
1
1
1
1
1
1
Z_1
Z_2
unused
unused

Figure 3.2 Example Solution

CELLULAR ARRAYS FOR PERFORMING CONVOLUTIONS.

RENATO DE MORI
CENS - IENGF Politecnico - TURIN -10129 Italy

ABSTRACT

Some cellular arrays for performing convolutions are introduced.These structures are constructed such that there is a large overlapping between the execution of multiplications and additions required to compute an output sample. Due to this overlapping, the time of a multiplication gives only an additive contribution to the overall time needed to compute an output sample. This leads to a remarkable improuvement in speed with respect to the previous presented solutions.

INTRODUCTION

A fast digital system for performing convolutions would make practical several good theories concerning digital signal transmission.

This paper introduces some cellular circuits for this purpose. They are constructed in such a way that the time of a multiplication gives only an additive contribution to the total time needed for computing an output sample.

In addition to the advantage of the high speed, they are designed as arrays of equal cells satisfying an important requirement for Large Scale Integration.

SPECIAL PURPOSE ITERATIVE ARRAY

A nonrecursive digital filter actuates a discrete convolution between a set of coefficients and a set of successive samples of a band-limited waveform X (t).

If A_0, ...,A_l, ...,A_L are the coefficients and X(nT-T), ..., X (nT-lT), ..., X (nT-LT), are the L samples preceding the n-th sample of the input signal X (nT), then the n-th sample of the filter output Y (nT) is given by:

$$Y(nT) = \sum_{l=0}^{L} A_l \, X(nT-lT) \qquad (1).$$

Assume that the input sample X (nT-lT) be represented by the 2's - complement binary fraction $\tilde{X}_{n,l}$; let N + 1 the number of bits of $\tilde{X}_{n,l}$ and $S_{n,l}$ its sign.

The coefficients are assumed to be represented in sign and magnitude. Let S_{A_l} and $\left|A_l\right| = \sum_{i=0}^{K-1} a_{i,l} 2^i$ be respectively the sign and the magnitude of the coefficient A_l.

In order to deal with fixed - point binary fractions having the advantages, for what overflows is concerning, remarked by Chu [1] and Jakson and al. [2], the coefficients are assumed to be all multiplied by 2^{-K}.

An output sample Y (nT) will be represented by an N + K + 1 bit - 2's - complement binary fraction $\tilde{Y}_n$ obtained as follows:

$$\tilde{Y}_n = \left| \sum_{l=0}^{L} \tilde{Y}_{n,l} \right|_2 \qquad (2);$$

where $\tilde{Y}_{n,l} = |A_l| \; X'_{n,l}$ and

$$X'_{n,l} = \begin{cases} \tilde{X}_{n,l} \, 2^{-K} & \text{if} \quad S_{n,l} \oplus S_{A_l} = 0 \\ (2-\tilde{X}_{n,l} \, 2^{-K}) & \text{if} \quad S_{n,l} \oplus S_{A_l} = 1. \end{cases} \qquad (3).$$

The relation (2) shows that a convolution is obtained by a sequence of modulo-two additions. The addends are products which can also be obtained by modulo-two additions of the bits of the fractions $X'_{n,l}$ shifted and controlled by the bits of $|A_l|$ (l = 0, 1, ..., L). The two different types of addions can overlap allowing a large saving in time. This saving can be improouved if the additions are performed with carry-save techniques.

A single array whose configuration is shown in fig.1a can be used for performing convolutions with an arbitrary number of coefficients. The cell structure is represented in fig. 1 b.

The array behaviour is timed by a clock.

During the first clock interval, the inputs to the array are given by the following equations:

$$A1 = |A_L| \quad ; \qquad I1 = X'_{n,L}$$

$$A2 = |A_{L-1}| \quad ; \qquad I2 = X'_{n,L-1} \quad ;$$

the array produces a set of partial addends whose sum is:

$$S\ (1) = \left| \, |A_L| \; X'_{n,L} + |A_{L-1}| \; X'_{n,L-1} \right|_2 \qquad (4);$$

the bits of these addends are stored into the flip-flops of each cell and the left-side external flip-flops.

The outputs σ of the full adders within cells of the same column can be seen as the bits of binary fractions, which, starting from the right-most column, can be indicated as:

$$s_0\ (1);\ s_1(1);\ ..\ ;\ s_i(1);\ ..\ ;\ s_{K-1}\ (1) \qquad (5);$$

analogously the carry outputs γ in the cells of a column can be thought as bits of the binary fractions:

$$c_0(1);\ c_1(1);\ \dots\ ;\ c_i(1);\ \dots\ ;\ c_{K-1}(1) \qquad (6)$$

where, while the least significant bit of $c_i(1)$ has weight $2^{-(N+K)+i+1}$, the least significant bit of $s_i(1)$ has weight $2^{-(N+K)+i}$ $(i= 0, 1, \dots, K-1)$.

The (4) can be also expressed as:

$$S(1) = \left| \sum_{i=0}^{K-1} \left\{ s_i(1) + c_i(1) \right\} \right|_2 \qquad (7)$$

Similar considerations can be done for the full adders of a column, whose outputs ϑ represent the set of binary fractions:

$$R_0(1);\ R_1(1);\ \dots\dots;R_i(1);\ \dots\dots;\ R_{K-1}(1) \qquad (8)$$

and whose outputs χ represent another set of fractions:

$$G_0(1);\ G_1(1);\ \dots\dots;\ G_i(1);\ \dots\dots;\ G_{K-1}(1) \qquad (9)$$

These numbers are related to those of sets (5)and (6)as follows:

$$\begin{array}{l} R_0(1) = s_0(1)\ ;\ \ G_0(1) = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \left| R_1(1) + G_1(1) \right|_2 = \left| c_0(1) + s_1(1) \right|_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \left| R_i(1) + G_i(1) \right|_2 = \left| c_{i-1}(1) + s_i(1) \right|_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \left| R_{K-1}(1) + G_{K+1}(1) \right|_2 = \left| c_{K-2}(1) + s_{K-1}(1) \right|_2 . \end{array} \qquad (10)$$

The numbers $R_0(1);\ \dots;R_i(1);\ \dots;\ R_{K-1}(1)$ and $G_0(1);\dots;G_i(1);\dots;G_{K-1}(1)$ are stored into the flip-flops inside the cells while into the column of flip-flops at the left-side of the array is stored the number $c_{K-1}(1)$.

Notice that the array performs modulo two additions by neglecting the carries having weight 2^1 or higher. Thus $S(1)$ can be obtained from the stored partial addends as follows:

$$S(1) = \left| \sum_{i=o}^{K-1} \left\{ R_i(1) + G_i(1) \right\} + c_{K-1}(1) \right|_2 \qquad (11)$$

During the successive clock intervals the array executes similar operations on the stored partial addends, the input samples and the corresponding coefficients up to the F-th interval $\left(F = \left\lceil \frac{L+1}{2} \right\rceil\right)$. (1)

These operations are summarised by the relations:

$$\left| s_i(f) + c_i(f) \right|_2 = \left| a_{i,L-2f} X'_{n,L-2f+2} + a_{i,L-2f+1} X'_{n,L-2f-1} + R_{i+1}(f-1) \right|_2 \tag{12}$$

$$c_{K-1}(f) = R_K(f)$$

$$\left| R_i(f) + G_i(f) \right|_2 = \left| c_{i-1}(f) + G_i(f-1) + s_i(f) \right|_2 \tag{13}$$

$(f=1, \ldots\ldots, F;\ i = 0, \ldots\ldots, K-1)$,

$R_i(0) = G_i(0) = 0.$

After the F-th clock interval, all the specification inputs (those of the input samples and of the coefficients) are inhibited and, during other N+K clock intervals, all the stored partial addends are added toghether to produce an output sample; the array operation is described by the following equations:

$$A_1 = A_2 = I_1 = I_2 = 0$$

$$\left| R_i(\tau) + G_i(\tau) \right|_2 = \left| G_i(\tau-1) + R_{i+1}(\tau-1) \right|_2$$

$$\tau = F+1, \ldots\ldots F+N+K; \quad i=0, \ldots\ldots, K-1. \tag{14}$$

It is easy to see from fig. 1a. that bits of the same weight are carried down from left to right by diagonal paths while the bits of the final result, which are produced sequentially from the least significant, are continuously restored into the cell of the right-most column till all the flip-flops are cleared.

During the last N+K clock intervals, other operations if required, such as serial 2's complementation, can be performed for the next sample computation outside of the Arithmetic Array.

The memory for the input samples and the coefficients can be easly designed with logic-in-memory arrays having provisions, when required, for serial 2's - complementation of the input samples.

The whole number N_{CL} of clock intervals needed for computing an output sample is:

---

(1) $\lceil V \rceil$ means V if it is integer; the immediately higher integer of V if it is not integer.

$$N_{CL} = F + K + N \qquad (15).$$

The clock period must be a slighty longer than the delay of two full adders and is of the order of 10 nsec using Schottky TTL microcircuits.

An optional command can be inserted for by-passing the cell flip-flops after the first F clock intervals making the array a purely combinational one.

In this case the last K+N clock intervals can be substituted by a single one of longher duration.

The number of cells for the array is

$$N_{1C} = (N+1) . K + \frac{K (K+1)}{2} \quad .$$

The proposed structure operates on two products of the convolution at a time; this number can be made higher, using more complex cells, resulting a reduction of the first addend of (15) and an increase of the clock interval.

The optimal number depends on the technology used for implementing the array.

AN ALTERNATIVE PARALLEL STRUCTURE

A large saving in the number of cells can be achieved using the cellular but not fully iterative array of fig.2.

The basic cell of the array is that of fig.1b.

The array refers to coefficients of 3 bits plus sign and variables of 3 bits plus sign.

The fourth row of the array adds corrective terms when the variables are negative and the last row performs the final addition to obtain the most significant bits of the output fraction $\tilde{Y}_n$.

The algorithm follows a suggestion by Deegan [3] for combinational multipliers.

A multiplication, when a variable $\tilde{X}_{n,l}$, is negative, is performed with the following algorithm.

Let $\tilde{X}_{n,l} = 2 - x_{n,l}$.

If $\tilde{Y}_{n,l}$ is the

fraction representing the product $A_l . X (nT-lT)$, its fractional part can be obtained as follows:

$$\tilde{Y}_{n,l} - 1 = \left|A_l\right| 2^{-K} (1 - x_{n,l}) + 1 -$$
$$- \left|A_l\right| 2^{-K}$$

This fraction is obtained multiplying $\left|A_l\right| 2^{-K}$ by the fractional part $1 - x_{n,l}$ of $\tilde{X}_{n,l}$ and adding the fractional part of the 2's-complement of $|A_1|\,2^{-K}$. This fractional part is obtained complementing all the bits of $|A_1|\,2^{-K}$ and adding 2^{-K}.

It is easy to see from fig.2 how the corrective terms are conditioned by the signs of the variables.

All the carries having weight 2^0 are added in the two left-most cells of the last row of fig. 2 with the sign bits of the variables for producing the sign bit of $\tilde{Y}_n$.

The proposed alternative solution does not change practically the operation time; the interconnaction paths is not fully iterative, but the number of cells is reduced to

$$N_{2c} = K.\,(N+2)+2\,.$$

ACKNOWLEDGEMENTS

This work was carried out at the "Centro di Elaborazione Numerale dei Segnali" of the National Council of Researches of Italy.

The Author is grateful to Profrs. R. Sartori and A.R. Meo for the useful discussions and suggestions.

REFERENCES

/1/ Chu Y. :"Digital Computer Design Fundamentals", Mc Grow Hill Ed., 1962 .

/2/ Jakson L.B., Kaiser J.F.,Mc Donald H.S.:" An approach to the implementation of digital filters." I.E.E.E. Trans., vol. AU16, No.3, pp.413÷421, Sept.1968.

/3/ Deegan, ID.: "Cellular multiplier for signed binary numbers". Electronics Letters, 1971, 7, pp.436÷437.

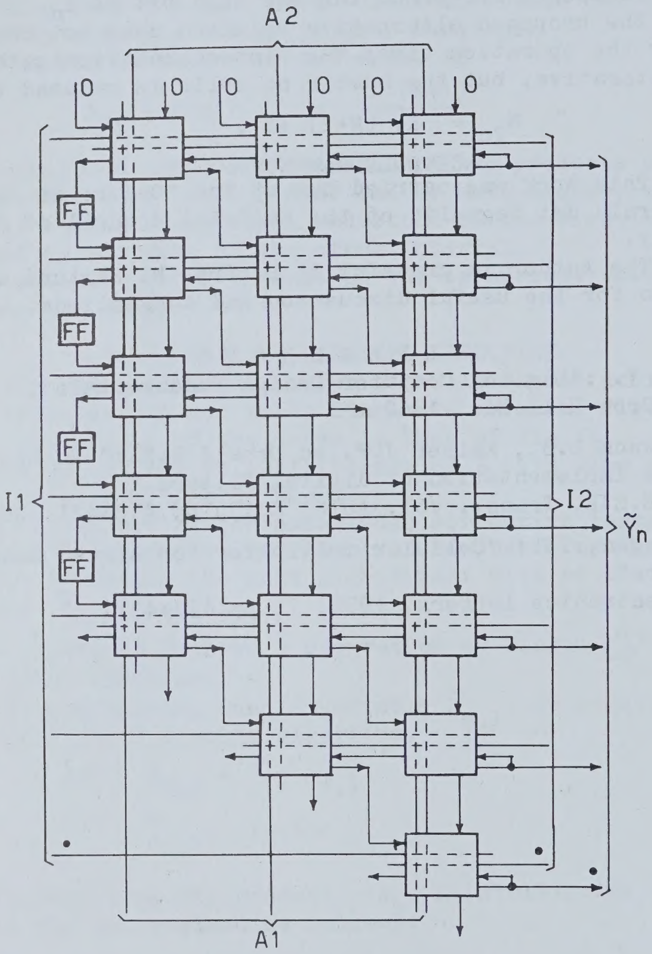

A2
0
0
0
0
0
0
FF
FF
FF
FF
I1
I2
$\tilde{Y}_n$
A1

FIG.1a

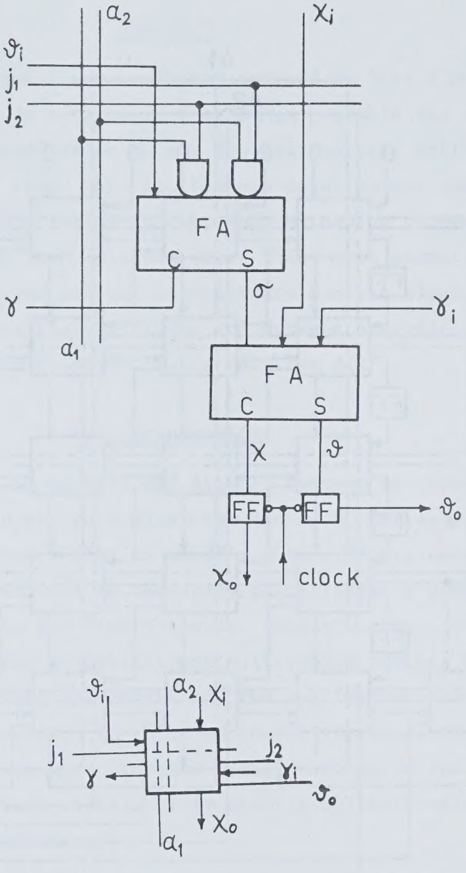

a2
xi
ϑi
j1
j2
F A
C
S
σ
γ
γi
a1
F A
C
S
χ
ϑ
FF
FF
ϑo
χo
clock
ϑi
a2
xi
j1
j2
γ
γi
ϑo
χo
a1

FIG 1b

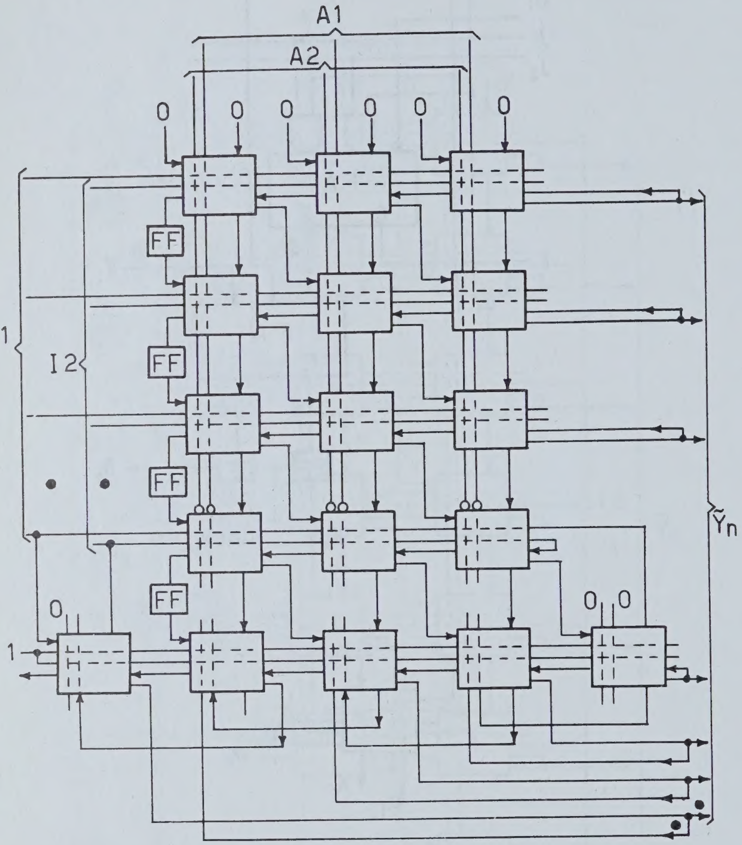

A1
A2
0
0
0
0
0
0
FF
FF
FF
FF
I1
I2
Ỹn
0
0
0
1

FIG. 2

THE RELATION OF EQUIVALENCE FOR POST ALGEBRAS

E. G. DUCASSE and G. A. METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

A certain equivalence relation defined on Post algebras of one-variable functions on an n-chain ($n \geq 2$) makes possible the static and dynamic hazard-free realization of any Boolean function defined on a Boolean lattice $B(2^n)$ [9]. This equivalence relation has now been generalized to arbitrary Post algebras, thus extending previous work carried out for finite such lattices [2]. These most recent results also offer hope that functions defined on arbitrary Boolean algebras might be implemented by circuits free of static and dynamic hazards in much the same way as functions defined on finite lattices $B(2^n)$.

I. INTRODUCTION

The concept of multivalued digital systems is indeed attractive. A computer having an n-ary ($n > 2$) arithmetic unit, for example, could do calculations faster than a similar machine with a binary unit, since a smaller number of bits would be necessary to represent a decimal number in an n-ary system than in the binary system. Unfortunately, with the present state of technology, the reliability of multivalued devices is not as good as similar binary systems. In addition, the multivalued circuits tend to be more expensive. However, there is no reason to assume that future technological advances cannot overcome these problems to the extent that the advantages of devices capable of processing multiple-valued signals will outweigh the disadvantages.

This paper is concerned with that aspect of multivalued switching theory called Post algebra. This algebra can serve as the mathematical foundation for this theory just as Boolean algebra does for binary switching theory. The particular aspect of Post algebras treated herein is an equivalence relation which can be defined for arbitrary Post lattices. More restricted versions of this relation have already proved extremely useful. The original formulation [9] was concerned only with one-variable functions defined on a Post chain $P(n)$; that is, an n-chain of two or more nodes. These functions themselves form a Post algebra

$G[P(n):x]$ which is isomorphic to $[P(n)]^n$, the direct product of n n-chains. The equivalence relation defined for these functions makes possible a static and dynamic hazard-free multivalued circuit implementation of each function defined on a Boolean algebra $B(2^n)$.

This original relation was later generalized to all finite Post lattices $[P(n)]^m$ [2]. This is significant because these lattices include each Post algebra $G[P(n):x_1,\ldots,x_k]$, the collection of all k-variable Post functions defined on the chain $P(n)$. The lattice $G[P(n):x_1,\ldots,x_k]$, in fact, is isomorphic to the direct product $[P(n)]^{n^k}$. This paper extends these earlier equivalence relations to arbitrary Post algebras, both finite and infinite. To conserve space, the proofs of the results presented here, many of which are quite lengthy, are omitted. They will appear in detail later [3].

II. FUNDAMENTAL CONCEPTS

In this section are presented the basic definitions and theorems on the relation of equivalence defined for arbitrary Post lattices. We begin by stating for reference the definition of Post algebra.

DEFINITION 1. Let n be an integer satisfying $n \geq 2$. A Post algebra P is a distributive lattice with zero element θ and universal element U for which the following conditions hold:

AXIOM 1. There exist n elements $e_o,\ldots,e_{n-1}$ in P satisfying

(1a) $\theta = e_o \leq e_1 \leq \cdots \leq e_{n-1} = U$;

(1b) if $x \in P$ and $x \cdot e_1 = \theta$, then $x = \theta$;

(1c) if $x \in P$ and $x + e_{i-1} = e_i$ for some i, then $x = e_i$.

AXIOM 2. For each element $x \in P$, there exist n elements $C_o(x),\ldots, C_{n-1}(x)$ in P satisfying

(2a) $C_i(x) \cdot C_j(x) = \theta$ if $i \neq j$;

(2b) $\sum_{i=0}^{n-1} C_i(x) = U$.

AXIOM 3. For each $x \in P$, $x = \sum_{i=0}^{n-1} (e_i \cdot C_i(x))$.

It is clear from this definition that Post algebra is an extension of Boolean algebra: the elements e_i are generalizations of the constants θ and U, and the functions C_i generalize the identity and complement functions of a Boolean algebra. Every Boolean algebra B is a Post algebra with $n = 2$, where $e_o = \theta$, $e_1 = U$, $C_o(x) = \overline{x}$, and $C_1(x) = x$ for all $x \in B$. Note that not every element of a Post algebra P need have a complement. Those elements of P which are complemented form a Boolean algebra, called the underlying Boolean algebra of P and denoted by P_B.

To introduce the relation of equivalence, we first define for an arbitrary Post algebra P a subset P_R of P to be the collection of all elements $r \in P$ having the property that if $r \leq b$ for some element $b \in P_B$, then $b = U$. This notion is illustrated in Figure 1, the lattice of all one-variable functions defined on the 3-element chain P(3). In this figure the 3-tuple abc represents the function $f(x) = x^{abc}$; that is, the function f which assumes the value a when x = 0, b when x = 1, and c when x = 2. The unit element for this lattice is the function 222. The set P_R then consists of the functions 111, 112, 121, 122, 211, 212, 221, and 222. It is not difficult to show that the lattice of Figure 1 is itself a Post algebra, namely, $[P(3)]^3$. This example will be used frequently in the sequel.

It is more than mere coincidence that the elements of $[P_R(3)]^3$ are precisely those members of $[P(3)]^3$ having all nonzero components. In fact, in any finite Post algebra $[P(n)]^m$, $[P_R(n)]^m$ consists of all those m-tuples, each of whose components is not zero. This is an important result because it can be used to demonstrate that the definition of equivalence which is given next is a generalization of a previous definition formulated only for finite Post lattices $[P(n)]^m$.

> DEFINITION 2. Let $x,y \in P$, a Post algebra. Then x is equivalent to y, denoted $x \sim y$, if and only if there exists an element $r \in P_R$ such that $x \cdot r = y \cdot r$.

Consider the functions x = 012 and y = 021 of $[P(3)]^3$. The element r = 111 is in $[P_R(3)]^3$ and $x \cdot r = 012 \cdot 111 = 011 = 021 \cdot 111 = y \cdot r$. Therefore by Definition 2, $x \sim y$. Notice that if r is taken to be 211, then $r \in [P_R(3)]^3$ and again, $x \cdot r = y \cdot r$. Thus the element r of Definition 2 is not always unique. Now consider the functions x = 012 and z = 002. Let $r = r_1 r_2 r_3 \in [P_R(3)]^3$ and assume $x \cdot r = z \cdot r$. Then it must be the case that $x_i \cdot r_i = z_i \cdot r_i$ for i = 1,2,3. In particular, $x_2 \cdot r_2 = z_2 \cdot r_2$; that is, $1 \cdot r_2 = 0 \cdot r_2 = 0$. Since r_2 cannot be zero, this is a contradiction. In other words, no such element r can exist and, hence, $x \not\sim z$. More generally, since $[P_R(n)]^m$ consists precisely of those m-tuples having all nonzero components, it follows that two elements $x,y \in [P(n)]^m$ are equivalent when for each i, $1 \leq i \leq m$, x_i and y_i are either both zero or both nonzero.

It is a simple exercise to show that the relation of Definition

2 is an equivalence relation. Also, using the characterization given earlier for $[P_R(n)]^m$, it can be established that this relation is an extension of the concept of equivalence for finite Post algebras [2]. This in turn implies that equivalence for arbitrary Post systems is a generalization of the original definition of equivalence for one-variable functions defined on a Post chain. The significance of this original notion of equivalence lies in the way it partitions $[P(n)]^n$, an isomorphic copy of the lattice of all one-variable functions defined on P(n). This partitioning is done in such a way that each equivalence class contains at least one function which has no operation hazards [9], a variety of temporary false output unique to multi-stable circuits. If one selects such an operation-hazard-free representative from each class, then it is possible to implement any Boolean function defined on $B(2^n)$ in a form free of static and dynamic hazards.

Turning now to the more general relation of Definition 2, we begin by investigating properties of the function C_o of Definition 1. It will be seen that this function plays a vital role in the development of results concerning the relation of equivalence for Post algebras. We have mentioned that not every element x of a Post algebra need have a complement; that is, an element $\overline{x}$ such that $x+\overline{x}=U$ and $x\cdot\overline{x}=\theta$. The element $\overline{x}$ may be described by saying it is the smallest element y such that $x+y = U$ and the largest element z satisfying $x\cdot z = \theta$. Although x may not be complemented, it is true that such an element z always exists for each element of every Post algebra. Such an element is called the pseudo-complement of x and, in arbitrary lattices, is denoted by x^*. It has been shown that in Post algebras, $x^* = C_o(x)$ [4]. Thus the function C_o is seen to play an important role in Post algebras even before the relation of equivalence is investigated. The following theorem summarizes some of the most important properties of this function. The notation C_o^2 is used for the function C_o composed with itself.

THEOREM 3. Let P be a Post algebra and let $x,y \in P$. Then

(1) $x \leq C_o^2(x)$,

(2) $C_o^3(x) = C_o(x)$,

(3) $x \leq y \Rightarrow C_o^2(x) \leq C_o^2(y)$,

(4a) $C_o(x+y) = C_o(x)\cdot C_o(y)$,

(4b) $C_o(x\cdot y) = C_o(x) + C_o(y)$,

(5) $C_o^2(x+y) = C_o^2(x) + C_o^2(y)$,

$$C_o^2(x \cdot y) = C_o^2(x) \cdot C_o^2(y),$$
$$C_o^2(x^*) = (C_o^2(x))^*.$$

These properties merit some discussion. Property (1) is a weakened form of the involution law of Boolean algebra. It may be the case that $x < C_o^2(x)$. For example, consider $[P(3)]^3$ and let $x = 210$. Then $C_o^2(x) = C_o(002) = 220$, which is strictly greater than x. Property (2) is a weak law of involution. If we twice take the pseudo-complement of the pseudo-complement of an element, $C_o^2(C_o(x))$, we recover the pseudo-complement of that element. From (2) it is not possible to infer $C_o^2(x) = x$, of which we have already seen a counterexample.

Property (3) shows that the function C_o^2 preserves order. An order-preserving function on a lattice is called an order-homomorphism. Equalities (4a) and (4b) are analogues for Post algebra of the familiar DeMorgan laws for Boolean algebra. Using these rules and the fact that $x^* = C_o(x)$, it is trivial to establish (5). A function such as C_o^2 which preserves addition, multiplication, and pseudo-complements is called a lattice-homomorphism. It is a simple matter to show that every lattice-homomorphism is an order-homomorphism, thus demonstrating that (5) is a stronger condition than (2).

III. CHARACTERIZATIONS OF EQUIVALENCE

In this section the equivalence relation of Definition 2 will be investigated in detail. This will be done through a study of its equivalence classes. In attempting to extend results from finite to arbitrary Post algebras, use can no longer be made of the simple representation theory available in the finite case. Indeed, it would be expected that many properties of the lattices $[P(n)]^m$ would not carry over to the general case. This is especially true of properties of least upper bounds and greatest lower bounds. Although infinite Post algebras P need not be complete, there is a very important class of least upper bounds which always exists.

THEOREM 4. Given $x \in P$, $C_o^2(x)$ is the least element of P_B which is at least as great as x.

Recalling Definition 2, we immediately obtain the following characterization of P_R.

COROLLARY 5. In any Post algebra P, $P_R = C_o^{-1}(\theta) = C_o^{-2}(U)$.

Thus the class P_R has been related to a standard part of Post algebras.

What is sought though is a characterization of the equivalence classes of the relation of Definition 2. To accomplish this we need two preliminary results. The first of these shows P_R to be precisely the set of sums of elements of P and their pseudo-complements.

LEMMA 6. For any Post algebra P, $P_R = \{x \in P | x = y + C_o(y)$ for some $y \in P\}$.

The second preliminary result needed for our next theorem gives a necessary condition for two elements of a Post algebra to be equivalent.

LEMMA 7. If $x,y \in P$ and $x \sim y$, then $C_o(x) = C_o(y)$.

Using Lemmas 6 and 7 it is now possible to characterize the equivalence classes of the relation of Definition 2 in terms of products $x \cdot P_R = \{x \cdot r | r \in P_R\}$.

THEOREM 8. For each $x \in P$, $[x] = C_o^2(x) \cdot P_R$, where $[x]$ is $\{y \in P | y \sim x\}$.

Now assume that $x \in P_B$. Then $C_o^2(x) = x$. Thus we have the following result.

COROLLARY 9. For each $b \in P_B$, $[b] = b \cdot P_R$.

The products $b \cdot P_R$ superficially resemble the familiar cosets of group and ring theory. However, there are several important differences. The cardinalities of these products are not generally equal. Furthermore, b is the only element of $b \cdot P_R$ which can be used as class representative. Since there is a unique $b \in P_B$ in each class $[x]$, each complemented element of P determines an equivalence class just as in the finite case [9]. To see how completely b specifies its equivalence class, we look next at an interval characterization of these classes.

To show that each class $[x]$ is an interval will require some preliminary work. The following weaker result, however, is easily established.

THEOREM 10. Each equivalence class of a Post algebra P is a sublattice of P.

If $x,y \in P$ and $x \leq y$, the interval $[x,y]$ is defined to be $\{z \in P | x \leq z \leq y\}$. We begin our quest for an interval characterization of the general equivalence class $[x]$ by giving such a representation of the set P_R.

LEMMA 11. In any Post algebra P, $P_R = [e_1, U]$.

The next result gives a distributivity property for elements over intervals. Although it is not shown, the proof of this lemma uses no special properties of a Post algebra beyond its distributivity. It is valid, therefore, for all distributive lattices.

LEMMA 12. If $x,y,z \in P$, then $x\cdot[y,z] = [x\cdot y, x\cdot z]$.

Using Lemmas 11 and 12, it is now easily shown that each equivalence class $[x]$ is an interval whose endpoints are determined by the element of P_B which it contains.

THEOREM 13. For each $x \in P$, $[x] = [C_o^2(x)\cdot e_1, C_o^2(x)]$.

Employing Theorem 13, it is possible to establish the converse of Lemma 7, leading to the following necessary and sufficient condition for two elements of a Post algebra P to be equivalent.

THEOREM 14. For $x,y \in P$, $x \sim y$ if and only if $C_o(x) = C_o(y)$.

This theorem generalizes to the infinite case the previously mentioned result that two elements of a finite Post algebra are equivalent if and only if they have zeros for precisely the same components.

It has already been shown in Corollary 5 that $P_R = C_o^{-1}(\theta)$. With Theorem 14, each class $[x]$ can be expressed as $C_o^{-1}(b)$ for some $b \in P_B$. In fact, $b = C_o(x)$.

COROLLARY 15. For each $x \in P$, $[x] = C_o^{-1}(C_o(x))$.

This section continues with two additional interval characterizations of the classes $[x]$. Although neither of these results illuminates the form of the equivalence classes as clearly as does Theorem 13, each of them is vital in enabling us to relate these classes to more well-known structures in a Post algebra. As a notational convenience in the statement of the first result, let A denote $\{y \in P | y \in P_B \text{ and } y < C_o^2(x)\}$, where x is a given element of P.

THEOREM 16. For each $x \in P$, $[x] = [\theta, C_o^2(x)] - \bigcup_{b\in A} [\theta,b]$.

The difference operator in this result is the relative complement. This theorem can be improved considerably as a computational result by restricting the set A to those elements of P_B which are covered by $C_o^2(x)$, but we will not pursue this.

The second result is similar to Theorem 16. In this case, however, the intervals involved all have right endpoint U. For this result let A' denote $\{y \in P | y \in P_B \text{ and } C_o^2(x) < y\}$.

THEOREM 17. For each $x \in P$, $[x] = [C_o^2(x) \cdot e_1, U] - \bigcup_{b \in A'} [b \cdot e_1, U]$.
Like A, the set A' can be restricted to those elements of P_B which cover $C_o^2(x)$.

The structures to which we wish to relate the classes $[x]$ are ideals and filters of the Post algebra. A nonempty subset I of a lattice L is said to be an ideal of L if $x, y \in I$ implies $x + y \in I$ and $x \in I$, $y \in L$ imply $x \cdot y \in I$. Thus if $x \in I$ and $y \leq x$, then $y \in I$. An ideal is called principal if it consists of all elements $y \in L$ satisfying $y \leq x$ for a given $x \in L$. Such an ideal is denoted by (x). Since $\theta \leq y$ for all elements y of a Post algebra P, it follows that $(x) = [\theta, x]$. Therefore Theorem 16 can be expressed in terms of ideals.

THEOREM 18. For each $x \in P$, $[x] = (C_o^2(x)) - \bigcup_{b \in A} (b)$.

Another way to characterize the classes $[x]$ of a Post algebra P is to use filters of P. A nonempty subset F of a lattice L is called a filter of L if $x, y \in F$ implies $x \cdot y \in F$, and $x \in F, y \in L$ imply $x + y \in F$. Thus if $x \in F$ and $x \leq y$, then $y \in F$. A filter is called principal if it consists of all elements $y \in L$ satisfying $x \leq y$ for a given $x \in L$. Such a filter is denoted by $\langle x \rangle$. Since $y \leq U$ for every $y \in P$, it follows that $\langle x \rangle = [x, U]$. Therefore we have the following reformulation of Lemma 11.

THEOREM 19. In any Post algebra P, $P_R = \langle e_1 \rangle$.

Employing Theorems 8 and 19 we immediately obtain the following result.

COROLLARY 20. For each $x \in P$, $[x] = C_o^2(x) \cdot \langle e_1 \rangle$.

Finally, we can also formulate Theorem 17 in terms of filters.

THEOREM 21. For each $x \in P$, $[x] = \langle C_o^2(y) \cdot e_1 \rangle - \bigcup_{b \in A'} \langle b \cdot e_1 \rangle$.

IV. CONCLUSION

A generalization of the relation of equivalence for Post algebras, an important part of the hazard-free implementation of functions defined on finite Boolean algebras by Post functions, has been presented. This generalization to arbitrary Post lattices offers hope that the methods of realizing static and dynamic hazard-free representations of functions defined on algebras $B(2^n)$ can be extended to functions on arbitrary Boolean algebras. This generalization has also been related to several standard structures of Post algebras to facilitate its investigation. Characterizations have been given in terms of products $b \cdot P_R$, the function C_o, intervals, and ideals and filters. It is thought that these results will provide a foundation for the development of methods extending to arbitrary Boolean functions techniques currently available only for functions on finite Boolean algebras.

REFERENCES

1. Birkhoff, G., Lattice Theory, American Mathematical Society Colloquium Publication 25, revised edition, New York, 1948.

2. DuCasse, E. and Metze, G., "A Generalized Equivalence Relation for Post Algebras," Conference Record of the 1972 Symposium on the Theory and Applications of Multiple-Valued Logic Design, Buffalo, New York, pp. 25-32, May, 1972.

3. DuCasse, E., Ph.D. thesis, Department of Mathematics, University of Illinois, Urbana, Illinois, to be published as a report of the Coordinated Science Laboratory, University of Illinois, Urbana, Illinois.

4. Epstein, G., "The Lattice Theory of Post Algebras," Transactions of the American Mathematical Society, Vol. 95, pp. 300-317, 1960.

5. Metze, G. A., "An Application of Multivalued Logic Systems to Circuits," Proceedings of a Symposium on Circuit Analysis, University of Illinois, Urbana, pp. 11-1 to 11-14, 1955.

6. Post, E., "Introduction to a General Theory of Elementary Propositions," American Journal of Mathematics, Vol. 43, pp. 163-185, 1921.

7. Rosenbloom, P. C., "Post Algebras, I. Postulates and General Theory," American Journal of Mathematics, Vol. 64, pp. 167-188, 1942.

8. Trazcyk, T., "Axioms and Some Properties of Post Algebras," Colloquium Mathematicum, Vol. 10, pp. 193-209, 1963.

9. Wojcik, A. S., "Relationships Between Post and Boolean Algebras with Application to Multivalued Switching Theory," Coordinated Science Laboratory Report R-512, University of Illinois, Urbana, 1971.

10. Wojcik, A. S. and Metze, G., "Some Relationships Between Post and Boolean Algebras," Conference Record of the 1971 Symposium on the Theory and Applications of Multiple-Valued Logic Design, Buffalo, New York, pp. 173-182, May, 1971.

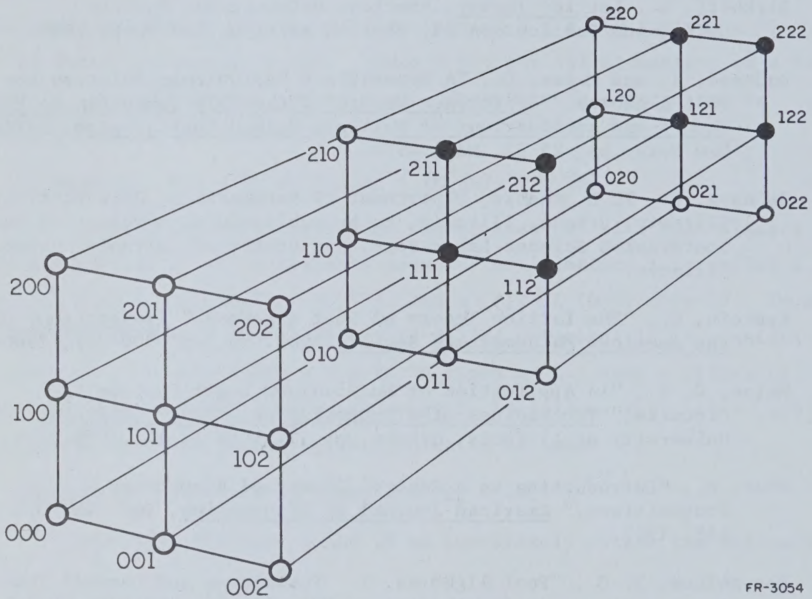

220
221
222
120
121
122
210
211
212
020
021
022
110
111
112
200
201
202
010
011
012
100
101
102
000
001
002
FR-3054

FIGURE 1

THE SET P_R IN $[P(3)]^3$

PRELIMINARY REPORT ON CONNECTIONS OF POST FUNCTIONS WITH SETTLEMENT OPERATORS

E. G. DUCASSE and G. A. METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

This investigation was motivated by the result $[x] = C_0^{-1}(C_0(x))$ for all elements x of a Post algebra [1]. This shows the settlement operator C_0 plays a central role in the theory of equivalence and, hence, in the hazard-free realization of a Boolean function [2]. It is only natural to inquire what role, if any, the other settlement operators $C_1,\ldots,C_{n-1}$ play in this theory. It will be demonstrated that each C_i, $1 \le i \le n-1$, can be used as an alternative to C_0, and that the functions $C_1,\ldots,C_{n-2}$ may actually serve more advantageously.

Since k-variable Post functions can be written as compositions of one-variable functions f(x) defined on n-chains P(n), the lattices of these functions, which are isomorphic to the direct products $[P(n)]^n$, will be stressed. However, since the results herein are valid for all finite Post algebras, these more general lattices will also be included. A considerable amount of information is already known about the sets $C_0^{-1}(b)$ for these finite algebras [1]. They are the equivalence classes of the relation of equivalence and thus partition each $[P(n)]^m$. Each set $C_0^{-1}(b)$ contains a unique member of $[P_B(n)]^m$, namely $\bar{b}$, and this element determines the set in the following sense: $C_0^{-1}(b) = [\bar{b}\cdot e_1, \bar{b}]$. Thus each $C_0^{-1}(b)$ is an interval and, consequently, is a sublattice of $[P(n)]^m$.

Due to the uniform structure of finite Post algebras, the form of each equivalence class is easily determined. If $b \neq U$, then $C_0^{-1}(b) \cong [C(n-1)]^k$, where k is the number of components of b which are zero and C(i) denotes the chain of length i; $C_0^{-1}(U) \cong C(1)$. It follows immediately that the cardinality of $C_0^{-1}(b)$ is $(n-1)^k$ for all $b \in [P_B(n)]^m$. It is also easily seen that the number of such classes isomorphic to $[C(n-1)]^k$ is $\binom{m}{k} = m!/k!(m-k)!$

The set of all one-variable functions defined on a Post chain P(n), $n \ge 2$, forms a Post algebra $[P(n)]^n$, under the MAX and MIN operations. Therefore these functions can be represented by n-tuples, each of whose components lies between 0 and n-1. Such a function f is said to be operation-hazard-free if whenever $0 \le x,y \le n-1$ and $|x-y| \le 1$, then $|f(x)-f(y)| \le 1$. Thus if $f = f_1\ldots f_n$ is the n-tuple representation of f, then f is operation-hazard-free if $|f_i - f_{i+1}| \le 1$ for all i = 1,...,n-2. We can extend this notion by defining $z = z_1\ldots z_m \in [P(n)]^m$ to be operation-hazard-free, denoted O-H-F, if $|z_i - z_{i+1}| \le 1$ for all i = 1,...,n-2. It is important that each $C_0^{-1}(b)$ contains an O-H-F element; in fact, $\bar{b}\cdot e_1$ is one. It is this result which makes possible the static and dynamic hazard-free realization of a Boolean function in terms of Post functions.

A set of O-H-F elements, precisely one of which comes from each $C_j^{-1}(b)$, will be called a complete set of O-H-F elements provided each element $x = x_1\ldots x_m$ satisfies $x_j = i$ or $x_j = i+1$ for all j = 1,...,m, where $0 \le i \le n-2$. Clearly only complete sets of Post functions should be employed in the realization of Boolean functions since any other set, even though it may consist solely of operation-hazard-free functions, will always contain at least two functions which when MAXed or MINed together give rise to operation hazards. It is a basic result that the $C_o^{-1}(b)$ contain a complete set of O-H-F elements; in fact, $\{b \cdot e_1 | b \in [P_B(n)]^m\}$, where $b \cdot e_1 \in C_o^{-1}(\bar{b})$, is such a set. It is significant that this set, which is the interval $[e_o, e_1] = [\theta, e_1]$, is the only complete set which can be obtained from these classes.

We now investigate the aforementioned properties with respect to arbitrary sets $C_j^{-1}(b)$, $j = 0, \ldots, n-1$. The relation of equivalence for Post algebras $[P(n)]^m$ can be formulated as follows: two elements $x = x_1 \ldots x_m$ and $y = y_1 \ldots y_m$ are equivalent when $x_i = 0$ if and only if $y_i = 0$ for each i. If we define x and y to be C_j-equivalent when $x_i = j$ if and only if $y_i = j$, then the equivalence classes of this relation are the sets $C_j^{-1}(b)$; in fact, $[x]_{c_j} = C_j^{-1}(C_j(x))$ for all $j = 0, \ldots, n-1$, where $[x]_{c_j}$ is the set of all elements C_j-equivalent to x. This generalizes the result for C_o.

The sets $C_j^{-1}(b)$, unlike $C_o^{-1}(b)$, are not generally intervals. They do, however, have the weaker property of being sublattices of $[P(n)]^m$. The only integer j, other than 0, for which $C_j^{-1}(b)$ is always an interval is $j = n-1$. In this case, $C_{n-1}^{-1}(b) = [b, b+e_{n-2}]$. The structure of the classes $C_j^{-1}(b)$, however, is the same as $C_o^{-1}(b)$. If $b \neq U$, then $C_j^{-1}(b) \cong [C(n-1)]^k$, k as before; $C_j^{-1}(U) \cong C(1)$. Thus $C_i^{-1}(b) \cong C_j^{-1}(b)$ for all $0 \leq i, j \leq n-1$. It also follows that the cardinality of $C_j^{-1}(b)$ is $(n-1)^k$. In addition, the number of such classes isomorphic to $[C(n-1)]^k$ is $\binom{m}{k}$.

Like $C_o^{-1}(b)$, each $C_j^{-1}(b)$ contains an O-H-F element. Moreover, for each $j = 0, \ldots, n-1$, the $C_j^{-1}(b)$ contain a complete set of O-H-F elements. The sets $C_{n-1}^{-1}(b)$, like the $C_o^{-1}(b)$, contain a unique complete set, namely $[e_{n-2}, e_{n-1}] = [e_{n-2}, U]$. On the other hand, the $C_j^{-1}(b)$, for each $j = 1, \ldots, n-2$, contain two such sets, these being $[e_{j-1}, e_j]$ and $[e_j, e_{j+1}]$. Thus the $C_j^{-1}(b)$, $1 \leq j \leq n-2$, provide a choice of O-H-F representatives.

To see that these C_j can be used as alternatives to C_o in the hazard-free realization of Boolean functions, consider the correspondence $b <\!\!-\!\!> b + e_j$. For $j = 1, \ldots, n-2$ (a more complicated rule can be used if $j = n-1$ is to be included), this is an isomorphism between the largest elements of the $C_o^{-1}(\bar{b})$, the elements used to represent Boolean functions [2], and the largest elements of the sets $C_j^{-1}(\bar{b})$. These elements of $C_j^{-1}(\bar{b})$ can then be replaced by O-H-F representatives.

REFERENCES

1. DuCasse, E., Ph.D. thesis, Department of Mathematics, University of Illinois, Urbana, Illinois, to appear as a report of the Coordinated Science Laboratory, University of Illinois, Urbana, Illinois.

2. Wojcik, A. S., "Relationships Between Post and Boolean Algebras with Application to Multivalued Switching Theory," Coordinated Science Laboratory Report R-512, University of Illinois, Urbana, 1971.

SYNTHESIS OF ASYNCHRONOUS CIRCUITS USING TRANSITION SENSITIVE FLIP-FLOPS

JON G. BREDESON AND PAUL T. HULINA
Department of Electrical Engineering
Pennsylvania State University
University Park, Pennsylvania

ABSTRACT

Recently, a good deal of research has been conducted on the use of transition or edge sensitive elements in the design of asynchronous sequential circuits. Transition sensitive memory elements have been used extensively in the implementation of synchronous sequential circuits as a means of eliminating the restrictions placed on pulse width. Several means are available for achieving multiple input or universal transition sensitive flip-flops. Here the approach is to employ a basic flip-flop as the memory element and add transition sensitive gating to the flip-flop input. In this manner, any type of transition sensitive flip-flop may be configured.

A procedure is then presented for synthesizing single change asynchronous sequential circuits directly from a primitive state table. For the single input change circuit, the number of internal variables needed to implement the reduced version of this table using conventionable flip-flops.

INTRODUCTION

In the recent past, a good deal of research has been conducted on the use of transition sensitive elements in the design of asynchronous sequential circuits (1, 2). Transition sensitive memory elements have been used extensively in the implementation of synchronous sequential circuits (pulse mode operation) as a means of eliminating the restrictions placed on pulse width (3, 4). One of the earlier papers dealing with transition sensitive devices in the design of asynchronous sequential circuits was due to Chuang (1). He has shown that these circuits (called transition logic circuits) can easily be used in the design of level-type asynchronous sequential circuits. Most recently, Smith and Roth (2) have proposed an inhibited toggle (I-T) flip-flop as a general purpose transition sensitive device. Here, an I-T flip-flop with a single T input is synthesized as an asynchronous circuit and multiple input I-T flip-flops are constructed using single T input I-T flip-flops with the outputs summed by an EXCLUSIVE-OR gate. It is interesting to note that Chuang's transition level multiplier when used in conjunction with a modulo two summer gives a realization almost identical with the I-T flip-flop proposed by Smith and Roth.

TRANSITION SENSITIVE GATE

In order to arrive at a mathematical expression describing the transition sensitive gate, some notation is needed. Consider a binary signal TS which is allowed to make only a single transition during a certain time interval. One of four possible events must occur, and they are:

1. Event (0, 1) - a 0 to 1 transition of TS.
2. Event (1, 0) - a 1 to 0 transition of TS.
3. Event (0, 0) - TS remains 0.
4. Event (1, 1) - TS remains 1.

The transition operators on the logic variable TS are defined as follows:

1. $\Delta TS = 1$ iff event (0, 1) occurs.
2. $\nabla TS = 1$ iff event (1, 0) occurs.
3. $\delta TS = \Delta TS + \nabla TS$ (+ denotes logical OR).

In addition we define an event which is associated only with the output x of a transition sensitive gate (TSG) as

5. Event (0, 1, 0) - a 0 to 1 to 0 transition of x.

The operator on the logic variable x is defined as follows:

4. $\rho x = 1$ iff event (0, 1, 0) occurs.

Shown in Figure 1 is a transition sensitive gate (TSG) with level input (L), transition sensitive input (TS) and output x. Here TS will be allowed to make only a single transition during a specific time interval. A verbal description of the operation of a transition sensitive gate is as follows:

1. The output x will be "1" for τ seconds following a transition on the transition sensitive input, TS, provided the level input L is maintained at "1" for some minimum time immediately preceding the transition.
2. The output x is "0" for all other input combinations or transitions on either input.

Mathematically a TSG with inputs TS and L, and output x can then be described in the following manner:

For TSG_{01} $\quad \rho x = \Delta TS \cdot L$ where $\rho x + x' = 1$ (1)

For TSG_{10} $\quad \rho x = \nabla TS \cdot L$ where $\rho x + x' = 1$ (2)

For TSG $\quad \rho x = \delta TS \cdot L$ where $\rho x + x' = 1$ (3)

The first expression in (1), (2), and (3) just describes the input-output relationship of the gate, while the second expression simply states that unless event (0, 1, 0) occurs, then x must be at a logic value 0.

A block diagram realization of a transition sensitive gate with delays is given in Figure 2. The normal OR operation of the output of two TSG's can be simplified as follows:

$$\rho(x_1 + x_2) = (x_2' \, \rho x_1 + x_1' \, \rho x_2 + \rho x_1 \, \rho x_2)$$

$$= \rho x_1[\rho x_2 + x_2'] + \rho x_2[\rho x_1 + x_1']$$

but since $x + x' = 1$, we have

$$\rho(x_1 + x_2) = \rho x_1 + \rho x_2 \tag{4}$$

It can be easily shown that the above result can be extended to give

$$\rho(x_1 + x_2 + \ldots, + x_n) + \rho x_1 + \rho x_2 + \ldots, + \rho x_n \tag{5}$$

Many commercial manufacturers have already produced the TSG_{10}.

TRANSITION SENSITIVE FLIP-FLOPS

The transition sensitive gate can be effectively used for input gating to a flip-flop composed of cross-coupled NOR gates to achieve any type of transition or edge sensitive flip-flop suitable for use in both synchronous and asynchronous sequential circuit design. Consider the circuit shown in Figure 3, with the assumption that the gates depected are of the TSG variety. Then, event δTS_{i1} when $L_{i1} = 1$ will set the flip-flop (i.e., $Q = 1$) if not already set. Similarly, event δTS_{i2} when $L_{i2} = 1$ will reset the flip-flop. Circuit operation will obviously be indeterminate if transitions occur simultaneously (within a specified time interval) on both TS_{i1} and TS_{j2} when $L_{i1} = L_{j2} = 1$ for any i and j. We now set out to mathematically describe the proposed model of the transition sensitive flip-flop. In the development to follow, we assume that the gates are strictly TSG. It is clear that in any of the results to be presented, that TSG_{01} or TSG_{10} gates may be used by simply replacing δTS by ΔTS or ∇TS respectively.

Our goal is to express the future state of the TSFF (Q^+) in terms of the present state (Q) and the inputs TS_{i1}, L_{i1}, TS_{i2}, and L_{i2}, where i = 1, 2, ..., m. To simplify matters, we write.

$$\rho A = \rho(A_1 + A_2 + \ldots, + A_m) \tag{6}$$

$$\rho B = \rho(B_1 + B_2 + \ldots, + B_m) \tag{7}$$

In addition, it is true that $\rho A \cdot \rho B = 0$ which indicates that the event (0, 1, 0) cannot be allowed to occur on both A and B within some specified interval of time since this would tend to both set and reset the flip-flop, causing indeterminate action. For the cross-coupled NOR gate flip-flop, we can write

$$Q^+ = \rho A + Q(\rho B)' \tag{8}$$

Substituting Eq. (6) and (7) into (8) and using the results obtained by (5) gives

$$Q^+ = \sum_{i=1}^{m} \rho A_i + Q\left[\sum_{i=1}^{m} (\rho B_i)\right]' \tag{9}$$

However, from (3) ($\rho x = \delta TS \cdot L$) and the TSG arrangement of Figure 3 we see that

$$Q^+ = \sum_{i=1}^{m} \delta TS_{i1} \cdot L_{i1} + Q\left[\sum_{i=1}^{m} (\delta TS_{i2} \cdot L_{i2})\right]' \tag{10}$$

$$Q^{+} = \sum_{i=1}^{m} \delta TS_{i1} \cdot L_{i1} + Q \prod_{i=1}^{m} [(\delta TS_{i2})' + L'_{i2}] \tag{11}$$

The above equation relates the next state behavior of the universal transition sensitive flip-flop to the inputs and present flip-flop status. It should be pointed out that any level input L_{ij} could be an AND gate before the L_{ij} of the TSG. The general model of an edge-sensitive flip-flop proposed by Smith and Roth can be easily constructed by using TSG_{10} and TSG_{01} gates with the transition sensitive inputs as class A inputs and the corresponding level inputs as class B inputs.

By appropriate connections, any type of transition sensitive flip-flop (TSFF) can be constructed. Consider the circuit shown in Figure 4 which is a delay flip-flop (D-TSFF). The D-TSFF shown here has m inputs but for the sake of simplicity, we can use only two inputs in the analysis to follow. Here we have:

$$\begin{aligned} TS_{i1} &= TS_{i2} = TS_i \\ L_{i1} &= D_i\, Q' \quad L_{i2} = D'_i\, Q \end{aligned} \qquad i = 1,\ 2 \tag{12}$$

Substituting Eq. (12) into the general TSFF Eq. (11) we get

$$Q^{+} = \sum_{i=1}^{2} \delta TS_i\, D_i\, Q' + Q \prod_{i=1}^{2} [(\delta TS_i)' + D_i\, Q']$$

$$Q^{+} = \delta TS_1 D_1\, Q' + \delta TS_2 D_2 Q' + Q[(\delta TS_1)' + D_1 + Q'][\delta TS_2)' + D_2 + Q'] \tag{13}$$

$$Q^{+} = (\delta TS_1 D_1 + \delta TS_2 D_2)Q' + [(\delta TS_1)'[(\delta TS_2)' + (\delta TS_1)'D_2 + (\delta TS_2)'D_1 + D_1 D_2]Q$$

In a specified time interval, four distinct events may occur.

a. $\delta TS_1 = \delta TS_2 = 0$ no transitions occur.
b. $\delta TS_1 = 1$ $\delta TS_2 = 0$ transition only on TS_1.
c. $\delta TS_1 = 0$ $\delta TS_2 = 1$ transition only on TS_2.
d. $\delta TS_1 = \delta TS_2 = 1$ transitions on both TS_1 and TS_2.

At this point, we examine the flip-flop behavior under these four conditions.

a. If in a specified time interval no transitions occur ($\delta TS_1 = \delta TS_2 = 0$), then Eq. (13) reduces to $Q^{+} = Q$ which indicates no change in the flip-flop status.

b. If a transition occurs only on TS_1, then Eq. (13) reduces to $Q^{+} = D_1$ which indicates that the value of D_1 at the time of a δTS_1 specifies the flip-flop status.

c. If a transition occurs only on TS_2, then the situation is identical to b and $Q^{+} = D_2$.

d. It is possible to tolerate transitions on both TS_1 and

TS_2 and still determine the flip-flop status if certain assumptions are imposed.

Consider restricting D_i such that it is not a function of any TS_i and also restricting the time difference (Δ_I) between transitions of TS_1 and TS_2 to be such that $\Delta_I \leq \Delta_F$ where Δ_F is the flip-flop delay from input to output. This insures that D_1 and D_2 will remain at fixed values until a transition occurs on TS_2 even through a transition occurred on TS_1 just prior ($\leq \Delta_I$) to the transition on TS_2. In this case Eq. (13) reduces to

$$Q^+ = (D_1 + D_2)Q' + D_1 D_2 Q \qquad (14)$$

This property of the flip-flop could allow us to process multiple input changes in asynchronous sequential circuits. Due to space limitations the multiple input change case has been omitted.

APPLICATION OF THE TSFF TO ASYNCHRONOUS SEQUENTIAL CIRCUITS

In this section, we consider the synthesis of fundamental mode asynchronous sequential machines using transition sensitive flip-flops, with the restriction that only single input changes are allowed. If during a specified time interval only a single transition occurs, then from previous discussion concerning flip-flop operation, we can write for the m input D-TSFF,

$$Q^+ = D_1\ \delta TS_1 + D_2\ \delta TS_2 + \ldots, + D_m\ \delta TS_m \qquad (15)$$

At this point, consider an m-input (x_1, x_2, ..., x_m) state table corresponding to an asynchronous machine. At a given stable state (s_k) we must, in general, be able to direct the machine to m different next states depending on the input change. We can conveniently describe this as

$$Q^+ = D_1\ \delta x_1 + D_2 + \delta x_2 + \ldots, + D_m\ \delta x_m \qquad (16)$$

Comparing Eq. (15) and (16) indicates that the number of TS inputs is equal to the number of machine inputs and $TS_i = x_i$ for $i = 1, 2, \ldots, m$ for each flip-flop. D_i is defined at the stable state of the machine as follows. Assume we are at stable state s_k and a change in x_i produces a state change to $s\ell$ which has the coding (a_1, a_2, ..., a_n). If D_{ij} is the ith input of flip-flop j, then $D_{ij} = a_j$ for $j = 1, 2, \ldots, n$).

As an example consider the minimal state machine represented by the state table of Figure 5(a). The state table and coded state table or transition table for a change in x_1 is shown in Figure 5(b). Note that a separate state table is needed for each input. We have thus transformed the asynchronous machine into a number of clocked synchronous machines, one for each input with the input being the clock for that particular synchronous machine. In Figure 5(c) is an implementation of the machine which will sequence properly without races because the final state is determined by the values on the D's at the time of a transition on a x imput. Since the feedback loop to all flip-flops is connected only during a transition on some x_i, no races can exist. Furthermore, no special factoring is required on the D inputs (level inputs) because transients do not effect the final state of each flip-flop after a transition occurs on a transition sensitive input. This is due to the characteristic of the builtin delay on the level input of the TSG. In addition it should be clear that any state assignment is allowed as in the case for synchronous machines.

In general, it is possible to reduce the number of feedback loops and thus the number of flip-flops by working directly with aprimitive state table instead of a minimal state representation of this primitive state table. The validity of the previous statement is based on the work reported by Singh (5) which deals with delayed input asynchronous machines. Singh has shown by use of partition theory that any normal primitive state table, corresponding to some asynchronous machine can be decomposed into two smaller machines. His results can be explained heuristically by considering a normal primitive state table with n states and m inputs $(x_1, x_2, \ldots, x_m)$. The state table will then have, in general, 2^m input columns which we denote as I_p for $p = 0, 1, 2, \ldots, 2^m - 1$. Further, let S_p denote the set of stable states in the pth input column and $|S_p|$ be the number of elements in S_p. Since the state table is primitive we must have $S_\alpha \cap S_\beta = \phi$ for $\alpha \neq \beta$. Thus, we can tell which group of states the machine is in simply by observing the inputs. Here we will need $\log_2 2^m = m$ state variables to distinguish between sets of states and, further, if all the states of S_p are given the same coding in the first m state variables as the input coding for I_p, then

$$y_i = x_i \quad \text{for } i = 1, 2, \ldots, m$$

It is now clear that the only memory devices (feedback) needed are for distinguishing between states in S_p under input column I_p. The number of state variables needed here will be governed by $S_{max} = \max(|S_0|, |S_1|, \ldots, |S_p|, \ldots, |S_{2^m-1}|)$. To separate the states we will need

$$k = \lceil \log_2 S_{max} \rceil$$

state variables or flip-flops.

As an example, consider the normal primitive state table of Figure 6(a) which is a 2 x 2 asynchronous unit delay (6). Here we will need 2 variables to distinguish between input columns and they are y_1 and y_2 as shown in Figure 6(b) with $y_1 = x_1$ and $y_2 = x_2$. To separate the states in each input column we require one variable since $|S_{max}| = 1$ and for this we use y_3. Figure 6(b) also gives the state table and coded state table for an x_1 input change. We notice that $D_{13} = D_{23} = y_2'$, however, we have no flip-flops for the variables y_1 and y_2. Since the D input to the flip-flop has a built-in delay, when a transition occurs on a TS_i, the flip-flop responds to the value on D_i just prior to the transition. Therefore, we may use $y_2 = x_2$ as a valid representation of the second state variable. Shown in Figure 6(c) is the circuit implementation which requires only one D-TSFF.

At this point, let us consider the desirability of implementing an asynchronous machine directly from the primitive state table. The state table of Figure 5(a) is a minimal version of the primitive state table which we have just implemented (Figure 7(a)). Here we have saved one flip-flop by implementing the machine directly from the primitive state table and this type of flip-flop reduction will be true in general for any machine. This statement can be explained by considering how a nonprimitive state table is transformed into a primitive state table.

Consider a nonprimitive state table with n states and $k (k \leq n)$ stable states in some input column I_j. We can validly assume that I_j contains more stable states than any other input column. This table will require $\lceil \log_2 n \rceil$ flip-flops. In order to make the table primitive, we must add more states to the machine so that there will only be one stable state per row. However, the number of stable states in the primitive state table will be identical to the number of stable states in the reduced table and furthermore, the number of stable states in input column I_j of the primitive state table will be identical to the number of stable states in I_j of the reduced state table. In order to implement the primitive state table we need only include a sufficient number of state variables to distinguish the k stable states in I_p which is $\lceil \log_2 k \rceil$. Thus the primitive version of the table will require fewer flip-flops unless $k > \frac{n}{2}$ in which case the flip-flops needed will be identical in both cases.

At this point, we present a brief step by step summary of the synthesis procedure.

1. Obtain a primitive state table representing the machine. Equivalent states (in the Huffman sense) should be removed. The only requirement is that the table be primitive ($S_\alpha \cap S_\beta = \phi$ for all α, β and $\alpha \neq B$). Call this table P.
2. Let P have m inputs $(x_1, x_2, \ldots, x_m)$, 2^m input columns denoted as I_p with $s \in S_p$ identical to the input coding for I_p with the first m state variables. This is done for each S_p where $p = 0, 1, 2, \ldots, 2^m - 1$. Then

   $$y_i = x_i \quad \text{for } i = 1, 2, \ldots, m$$

3. Let $S_{max} = \max (|S_0|, |S_1|, \ldots, |S_{2^m-1}|) = k$, then $|\log_2 k| = h$ variables will be needed, which we denote as $m + 1$, $m + 2$, $\ldots$, $m + h$ where $m + h \leq n$. Any coding can be chosen as long as all the states in each S_p are separated.
4. Draw individual state tables and coded state tables for each input transition x_i where $i = 1, 2, \ldots, m$. D_{ij} is determined from the individual tables for x_i. D_{ij} at each state is equal to the entry of the coded state table for $j = m + 1, \ldots, m + h$. If D_{ij} is a function of $y_1, \ldots, y_m$, use $x_1, \ldots, x_m$ respectively for these variables.

CONCLUSION

This paper has presented a generalized method for constructing transition sensitive flip-flops by use of transition sensitive input gating. As has been shown, these flip-flops can be used in the implementation of both single and multiple input change asynchronous sequential circuits. In either case, the state reduction problem is eliminated since one works with the primitive state table. In addition, the necessity to remove races and hazards is eliminated, since the circuits synthesized are race and hazard free.

REFERENCES

1. Y. H. Chuang, "Transition Logic Circuits and a Synthesis Method," IEEE Trans. on Computers, Vol. C-18, No. 2, February 1969, pp. 154 - 168.
2. J. R. Smith, Jr. and C. H. Roth, Jr., "Analysis and Synthesis of Asynchronous Sequential Networks Using Edge-Sensitive Flip-Flops," IEEE Trans. on Computers, Vol. C-20, No. 8, August 1971, pp. 847 - 855.
3. E. J. McCluskey, Introduction to the Theory of Switching Circuits, New York: McGraw-Hill, 1965.
4. F. J. Hill and G. R. Peterson, Introduction to Switching Theory and Logical Design, New York: John Wiley and Sons, Inc., 1968.
5. S. Singh, "On Delayed-Input Asynchronous Unit Delays," IEEE Trans. on Computers, Vol. C-19, No. 10, October 1970, pp. 896 - 902.

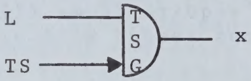
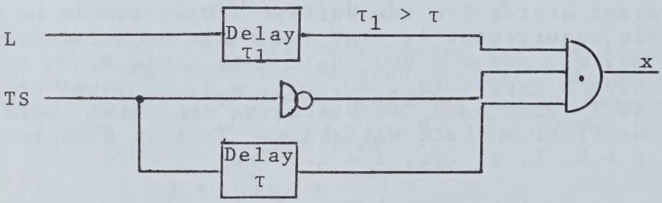
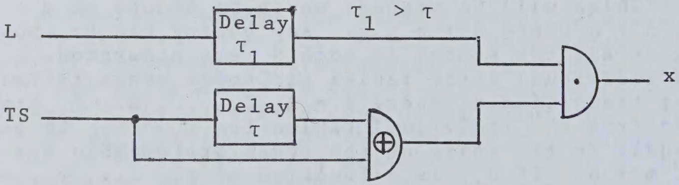


Figure 1. Transition Sensitive Gate



a. TSG_{10} Block Diagram



b. TSG Block Diagram

Figure 2

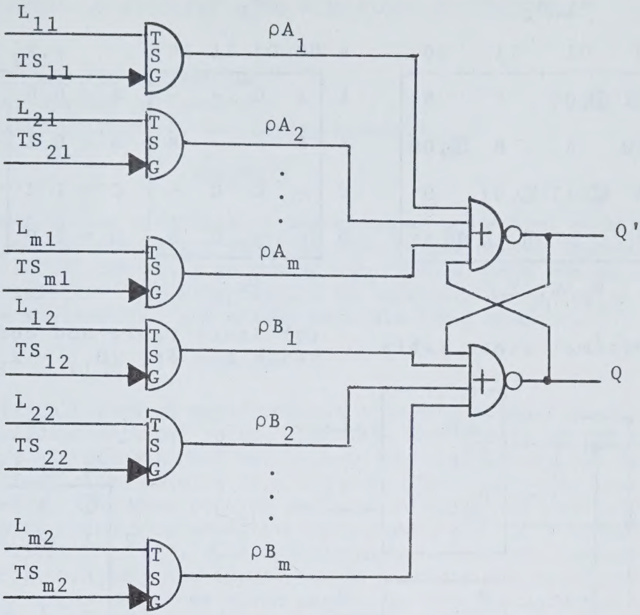

L_{11}
TS_{11}
L_{21}
TS_{21}
L_{m1}
TS_{m1}
L_{12}
TS_{12}
L_{22}
TS_{22}
L_{m2}
TS_{m2}
TSG
ρA_1
ρA_2
ρA_m
ρB_1
ρB_2
ρB_m
Q'
Q

Figure 3. Model of a Universal Transition Sensitive Flip-Flop

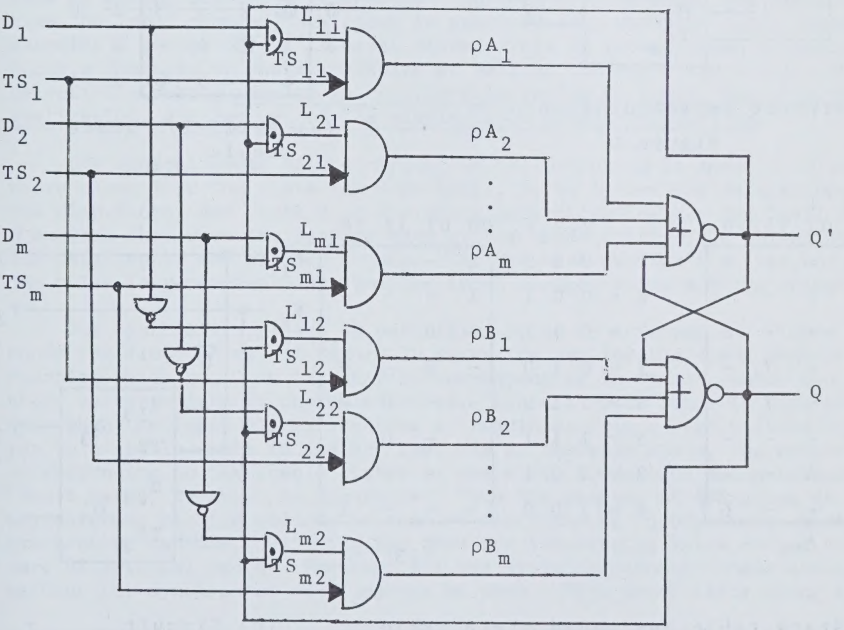

D_1
TS_1
D_2
TS_2
D_m
TS_m
L_{11}
TS_{11}
L_{21}
TS_{21}
L_{m1}
TS_{m1}
L_{12}
TS_{12}
L_{22}
TS_{22}
L_{m2}
TS_{m2}
ρA_1
ρA_2
ρA_m
ρB_1
ρB_2
ρB_m
Q'
Q

Figure 4. Model of a Delay Transition Sensitive Flip-Flop (D-TSFF)

s	$x_1 x_2$ 00	01	11	10
A	Ⓐ, 01	Ⓐ, 00	C	B
B	Ⓑ, 10	A	D	Ⓑ, 00
C	A	Ⓒ, 11	Ⓒ, 01	D
D	B	C	Ⓓ, 10	Ⓓ, 11

S, $Z_1 Z_2$

(a) Minimal state table

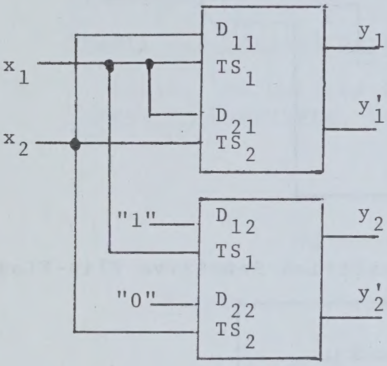
s	$x_1 x_2$ 00	01	11	10
A	B	C	-	-
B	B	-	-	B
C	-	C	C	-
D	-	-	C	B

δx_1

$y_1 y_2$	$x_1 x_2$ 00	01	11	10
A = 0 0	01	11	--	--
B = 0 1	01	--	--	01
C = 1 1	--	11	11	--
D = 1 0	--	--	11	01

D_{11} D_{12}

(b) State table and coded state table for δx_1 ($D_{11} = x_2$, $D_{12} = 1$)



x_1
x_2
D_{11}
TS_1
D_{21}
TS_2
y_1
y_1'
"1"
"0"
D_{12}
TS_1
D_{22}
TS_2
y_2
y_2'

(c) Circuit implementation

Figure 5

s	$x_1 x_2$ 00	01	11	10	Z_1	Z_2
1	①	2	-	3	0	1
2	1	②	4	-	0	0
3	6	-	5	③	0	0
4	-	7	④	8	0	1
5	-	7	⑤	8	1	0
6	⑥	2	-	3	1	0
7	1	⑦	4	-	1	1
8	6	-	5	⑧	1	1

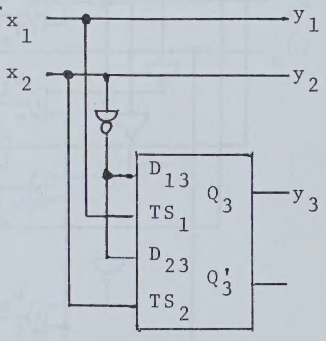
(a) Primitive state table

s	$x_1 x_2$ 00	01	11	10
1	3	-	-	-
2	-	4	-	-
3	-	-	-	6
4	-	-	7	-
5	-	-	7	-
6	3	-	-	-
7	-	4	-	-
8	-	-	-	6

δx_1

$y_1 y_2 y_3$	00	01	11	10
1 = 0 0 0	1	-	-	-
6 = 0 0 1	1	-	-	-
2 = 0 1 1	-	0	-	-
7 = 0 1 0	-	0	-	-
4 = 1 1 0	-	-	0	-
5 = 1 1 1	-	-	0	-
3 = 1 0 1	-	-	-	1
8 = 1 0 0	-	-	-	1

D_{13}



x_1
x_2
y_1
y_2
D_{13}
TS_1
D_{23}
TS_2
Q_3
Q_3'
y_3

(b) State table and coded state table for δx_1 ($D_{13} = y_2'$)

(c) Circuit implementation

Figure 6

SYNTHESIS OF ASYNCHRONOUS MACHINES USING FLIP-FLOPS AND CONTROLLED EXCITATION *

YING HUANG CHUANG AND SANTANU DAS
Computer Systems Laboratory
Washington University, St. Louis, Missouri

ABSTRACT

A new approach to the synthesis of asynchronous sequential machines is proposed. The internal states are realized with flip-flops, triggered with a clock pulse which is internally generated only when a state change is necessary. This selective clocking results in considerable saving in logic and more flexible realization, and allows multiple input changes.

SUMMARY

The most difficult part of asynchronous sequential circuit design is the problem of avoiding critical races. This has been mostly solved through critical-race-free state assignment which, however, complicates the design procedure and in most cases results in a lot more state variables than absolutely necessary. The race problem can also be solved by realizing the state variables with transition sensitive flip-flops [1,2] or with clocked (edge-triggered) flip-flops [3]. The underlying principle of such synthesis using flip-flops is that only the input changes provides the requisite triggering for state change. Once the clock passes the triggering threshold of the flip-flop, any change in the data input cannot affect the flip-flop, and thus the critical race is avoided. In this note, we propose to extend this idea by adding selective triggering. Instead of generating the clock each time the input changes, the clock is produced only when the input change warrants a change of the internal state. This is accomplished by making the clock a function of state variables as well as the input variables. This selective clocking results in considerable saving in logic, more flexible realization, and removal of the single input change restriction.

The general model of the circuit we are proposing is shown in figure 1, where block 1 is the state function logic, D_1 to D_m are the data inputs to the flip-flops, and block 3 is the clock generating logic. The small delay Δ ensures that when the leading edge of the clock reaches the flip-flops, the data inputs are already steady. The one-shot ensures that one and only one pulse is generated each time the input change causes a state change.

Our synthesis approach is best illustrated by an example. Figure 2a shows the flow-table, and figure 2b shows the map for the clock generating function which has 0 in the entries corresponding to stable states and 1 in those corresponding to unstable states. Since a clock pulse is generated only when the input change requires a transition from a stable state in one row to a stable state in another row, via an unstable state, the entries corresponding to the stable states in the state transition table become "don't cares" as shown in figure 2c. Thus the problem of designing an asynchronous machine reduces to that of designing an incompletely specified synchronous machine, obviating the need for introducing extra states to take care of critical races. Further, all the avalable economic state assignment methods for synchronous machines can be used. This possibility along with

---

* This work was supported by the Advanced Research Projects Agency of the Department of Defence under Contract SD-302 and by the Division of Research Resourses of the National Institute of Health under Grant RR-00396.

the large number of "don't cares" generally result in an economy of realization unmatched by any other existing synthesis methods.

The only operating constraint of our realization is that the rate of input change should be slower than the flip-flop switching time. It should be noted that when multiple input change is allowed, not all the stable state entries in the state transition table can be made "don't cares" and thus the realization is likely to be more costly than when only single input change is allowed.

REFERENCES

1. Y. H. Chuang, "Transition Logic Circuits and a Synthesis Method", IEEE Trans. Computers, Vol. c-18, pp. 154-168, February, 1969.

2. M. D. McIntosh and B. L. Weinberg, "On Asynchronous Machines with Flip-Flops", IEEE Trans. Computers, Vol. c-18, P. 473, May 1969.

3. J. G. Bredson and Paul T. Hulina, "Generation of a Clock Pulse for Asynchronous Sequential Machines to Eliminate Critical Races", IEEE Trans. Computers, Vol. c-20, pp. 225-226, February 1971.

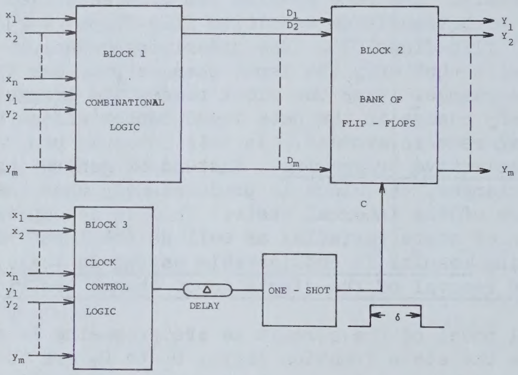

x1
x2
xn
y1
y2
ym
BLOCK 1
COMBINATIONAL
LOGIC
D1
D2
Dm
BLOCK 2
BANK OF
FLIP - FLOPS
Y1
Y2
Ym
C
x1
x2
xn
y1
y2
ym
BLOCK 3
CLOCK
CONTROL
LOGIC
Δ
DELAY
ONE - SHOT
δ

FIGURE 1: Model of an Asynchronous Sequential Machine Using Flip-Flop and Controlled Excitation

x_1x_2 00	01	11	10
Ⓐ	C	Ⓐ	D
Ⓑ	D	Ⓑ	C
B	Ⓒ	A	Ⓒ
A	Ⓓ	B	Ⓓ

(a) Merged Flow Table

STATE \ INPUT	00	01	11	10
A	0	1	0	1
B	0	1	0	1
C	1	0	1	0
D	1	0	1	0

(b) Map of the Clock Function

STATE \ INPUT x_1x_2	00	01	11	10
A	-	C	-	D
B	-	D	-	C
C	B	-	A	-
D	A	-	B	-

(c) State Transition Table

Fig. 2 Example Asynchronous Machine Illustrating the Synthesis Steps

ON COMPLETELY AND INCOMPLETELY SPECIFIED DIAGNOSABLE MACHINES

S. C. LEE
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF HOUSTON
HOUSTON, TEXAS 77004

SUMMARY

One of the important problems in designing sequential circuits, such as flip-flop circuits, is to make them to possess one or more distinguishing sequences thereby permitting us to identify uniquely the states of the machine by inspecting its response to such a sequence. The only work so far that has been published in the literature on this problem was by Kohavi.* He presented a method of designing diagnosable sequential circuits by adding to them an output terminal and assigning different output symbols to selected transitions. He has proven the following general result: To every reduced machine M there corresponds a definitely diagnosable machine M', which is obtained from M by the addition of one or more output terminals.

In this paper, an alternative approach to the problem is proposed. Instead of adding one or more output terminals, we add one or more input terminals to the machine and proves that every reduced machine can be made diagnosable by this technique. The two approaches are compared and discussed.

The problem of designing an incompletely specified machine has never been explored, yet it is so important that it is inevitable encountered in the design of flip-flop circuits of a n-state machine, where $n = 2^k$, k is a integer. In this paper, several methods of designing incompletely specified diagnosable machines are presented. Examples for illustrating the methods as well as demonstrating their usefulness in designing practical sequential circuits are given.

---

*Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, 1970, pp. 425-432.

OPTIMAL SYSTEM DESIGN USING DIODE ARRAYS AS LOGIC STRUCTURES AND READ-ONLY MEMORIES

ALLEN D. SYPHERD
North American Rockwell Microelectronics Company
Anaheim, California

ABSTRACT

Use of a large integrated Silicon-on-Sapphire diode array in system design is discussed. Techniques are presented which optimize system design by improving performance and reducing some system requirements. Comparisons are made to standard read-only memory and TTL logic systems which show a speed, parts count, and organizational advantage of the SOS device. Two examples of actual system designs are presented.

INTRODUCTION

A large scale integrated Silicon-on-Sapphire diode array offers unique characteristics which promote both design of optimal systems and optimal design of systems. Since this may appear to be doubletalk, it should be clarified that two processes are discussed. Design of optimal systems relates to that effort which yields on optimal system; one which obtains highest system performance in terms of system speed, accuracy, and data handling capacity. Optimal design relates to those factors which lower cost, improve reliability, and minimize engineering effort. This can be accomplished by reducing parts count and part types, lowering power requirements, reducing system design time with standardized circuit elements, and utilizing a single part type for multifunctional use in the system.

Silicon-on-Sapphire Diode Array

The SOS diode array is basically a large, high speed AND-OR gate with a maximum amount of organizational flexibility. Thus, it can be used as a logic structure implementing many large combinational logic equations, or a highly flexible, large capacity, read-only memory for table lookup, code conversion, character generation, or numerous other applications in logic systems. Rather startling reductions in cost and parts count can be realized using the array. The flexibility offered by the array permits logic minimization techniques to further reduce size and cost of systems.

Silicon on Sapphire has several qualities which make it ideally suited for diode arrays. The substrate is sapphire, an insulator, rather than the more common bulk silicon, and this eliminates parasitic capacitance to the substrate. Processing techniques produce vertical junctions which have extremely small cross-sections: 0.4 mil by 0.6 micron giving an individual junction capacitance in the range of 10 femtofarads (10^{-14} fd). These two factors combine to make a high speed device, and one in which large numbers of diodes can be connected together without building up capacitance. The diodes are regular pn junction devices. The interface is directly to TTL except on the outputs which are current sources and need a simple buffer sensor. The present device contains 128 resistors and 5120 diodes in a 40 x 128 matrix, yet a typical delay is only 20 nanoseconds between input and an output. Larger, more compact diode arrays are being designed which contain over 13,000 diodes on a single substrate, bringing even greater potential to system design using diode arrays.

How it Works

The SOS diode array has 128 rows which are attached through on chip resistors to a power bus as in Figure 1. The 40 columns, which are electrically identical, are brought out to package leads and can be assigned one of two functions.

Output An output column has diodes connected to only those rows which have data relative to that output. The 128 rows are divided into row groups, with all the rows of one row group feeding the same output column. There can be from 1 up to 32 outputs. Realistically, most systems use a maximum of about 16 outputs, with 9 outputs being about average.

Control The remaining columns perform several types of control functions. Row decoder columns act to select or unclamp a row or set of rows of the array. Two row decoder columns are required for each input variable since the input and its complement must be considered to uniquely select a row. In this manner the 128 rows may be divided into row groups which will feed different outputs. Other columns may act as enables and do not require the complement of the input variable. External decoders (such as a 1 of 16 decoder) may be connected to these columns. The external decoder will clamp one selected column at a time, and those rows connected to it will also be clamped. If a row is selected by the row decoder and there is no diode connected to the clamped enable column, then this row may go to a "high" state and a logic "one" will appear on the output column attached to that row.

There are two simple constraints on the organization of these column functions. If M outputs are required, the array must be divided into M row groups. Let N_m be the number of rows in the mth row group. Then,

$$\sum_{m=1}^{M} N_m \leq 128. \tag{1}$$

A special case, usually found in read-only memory applications, is $N_m = C$ for all m. If K columns are used for control and M columns are outputs,

$$K + M \leq 40. \tag{2}$$

Logic Applications

Diode arrays are an efficient way to mechanize logic equations. The diode array can be viewed as a large logical AND-OR gate. The voltage on each row is the AND of all the control signals which are connected through diodes to that row. The voltage of an output column is the OR of the voltages on all rows which have diodes connected to that column. Figure 2 shows the function $F = AB + \overline{A}\overline{B}\overline{C} + AC$ mechanized by 10 diodes and 3 resistors. Figure 3 shows the same function F mechanized with standard integrated circuit gates.

Comparison of the two techniques shows the speed advantage that diode arrays can have over logic gate mechanizations of complex logic equations. The diode array computes the AND-OR functions in one delay time; (the time necessary to charge up capacitance loading on the output through resistor R, which is 20 nsec). The logic gate approach for this simple example takes two levels of logic and the delays are additive. It is obvious that for very complex combinational logic expressions, several levels of gate logic with the inherent delays would be required while the diode array technique remains at one delay time. System speed performance can be optimized by using the diode array to reduce excessive propagation delays associated with multilevel logic. Parts count is reduced from several gate packages to one integrated array with standardized interfaces.

A previous study$^{(1)}$ has shown the advantage of using programmable read-only memories to replace random logic. The example indicated that one 64 bit programmable ROM could easily generate 4 functions of 4 variables and replace 17 gates with 50 inputs (approximately ten 14 pin dual in-line packages). The SOS diode array would require 15 rows by 12 columns to generate these same functions, or about 3.5% of its total logic capability. One array can compute the value of a logic function made up of up to 128 minterms spanning 19 input variables, an impressive amount of logic available on one chip.

Read-Only Memory Applications

Perhaps the more common use attributed to the large diode array is read-only memory storage.[2,3,4] Again, it is well suited to this application because it not only offers high speed, 20 nsec access, and high storage capacity, but a freedom of organization that allows the system designer to optimize the memory system to his particular needs. A number of systems have been designed using non-standard word lengths including: 5 bit words, 7 bit words, and 9 bit words. Now, when the system designer finds that the Optimum system configuration requires a 5 bit word from memory, he does not have to buy an 8 bit memory and throw 3 bits away, or compromise the design by making do with a 4 bit organization. This is an important feature which can lead to more efficient usage of ROM's in system design.

When used as a read-only memory, some of the diode positions must be used for outputs and decoder functions rather than data storage. Even with this requirement the array has a large capacity which can be organized in a variety of configurations, including: 1664 two-bit words, 832 four-bit words, 625 five-bit words, or 207 nine-bit words. It is the unstructured characteristic of the array which allows versatility of design. As an example, suppose it is required to store 768 five-bit words for a system. Further, suppose that within this 768 words there are one or more blocks of "zeros" with a total of 143 words for which the output is zero. Even though 768 five-bit words exceeds the capacity of the diode array, by simply not storing the values of address row decoders for those zero words, the problem will fit one array. Since the user has control over what addresses will be used, and is not confined to a predetermined block of addresses (e.g. word 0 to word 511), advantage may be taken of situations where blocks of data are zero or where blocks of data are identical.

Read-Only Memory by Logic

Another optimization process which increases the apparent bit capacity of the diode array is the logic structure approach rather than the straight forward read-only memory storage. A change in viewpoint reverses the standard question of what outputs should be high for each input word to ask: For each bit of the output word, what subset of input addresses is required? Each bit position of the output word will have a list of addresses corresponding to inputs for which that particular output bit is true. The expression for each output bit will be the OR of all the addresses (i.e., classical minterms) which are the AND functions of the input data. This technique not only eliminates storage of blocks of "zero" states, but eliminates storing zero values of bit positions within an output word, further reducing memory requirements. An example of this technique demonstrates the potential for very large reductions in storage requirements.

Example: Input Variables / Output Functions

A	B	C	F1	F2	F3	
0	0	0	0	0	0	
0	0	1	1	1	1	
0	1	0	1	0	1	
0	1	1	1	1	0	(3)
1	0	0	0	0	1	
1	0	1	0	0	1	
1	1	0	1	0	0	
1	1	1	0	1	1	

This truth table could be stored as a straight-forward read-only memory in a diode array. It would require 9 columns x 24 rows. Using the logic approach, a list of required addresses is made for each output bit. This bit can be further minimized to take advantage of "don't care" conditions in the address minterms.

	A B C		A B C		A B C	
F1	0 0 1	F2	0 0 1	F3	0 0 1	
	0 1 0		0 1 1		0 1 0	
	0 1 1		1 1 1		1 0 0	(4)
	1 1 0				1 0 1	
					1 1 1	

By inspection, the three functions can be reduced to the following: (x = don't care).

A B C	A B C	A B C	
0 X 1	0 X 1	X 0 1	
0 1 X	X 1 1	0 1 0	(5)
X 1 0		1 0 X	
		1 X 1	

A diode array layout of these functions is shown in Figure 4. "Don't care" conditions are implemented by removing both diode connections (variable and complement) from the input. This set of functions mechanized in a diode array requires 9 columns x 9 rows, almost a factor of 3 smaller than the read-only memory approach.

Computer Aid in Optimizing Storage

This simple example given could be reduced to a minimized set of minterms by inspection. For more realistic complex system design, a Quine-McClusky-type minimization should be performed on each subset of addresses. A computer program has been written to perform this minimization which will handle up to 10 input variables. Increasing the computer's capability much beyond this level creates an exponential rise in computer time which is probably not cost effective. The user must decide whether further reduction of the minterm list is worth a potentially costly computer run. The example showed a reduction in each minterm list from the minimization by inspection. F1 was reduced 25% (4 terms to 3), F2 - 33%, and F3 - 20%. In actual applications, computer minimization has given reductions in the minterm count of about 30%. This means that an optimized diode array will handle almost 1/3 more data than conventional storage techniques allow. Again, system size and cost are reduced.

System Design Examples

Control of sequential networks or other system functions is another logic application of the diode array. One system built using this control technique performs testing operations during the laser encoding[5] of new diode arrays. In this system, the diode array is used as a multifunctional logic structure. It stores set/reset equations for the sequential network state flip-flops, counter enables and timing controls, data latch enables, and an instruction list which is fed to the laser encoder. (A block diagram of the system is shown in Figure 5). Fifteen outputs from the array were required to perform these functions. The outputs are functions of 12 input variables and one enable. The stored data was optimized to take advantage of numerous "don't care" conditions for various illegal states and repeating data patterns. The diode array encoded to perform this control function is about half used (66 rows out of 128). To perform this same task with TTL logic gates and conventional ROM's would have required 88 words of 8 bit length in ROM's plus 48 gates with 140 inputs (approximately 16 standard 14 pin TTL packages). If only ROM storage were used as discussed in reference 1, the required storage would be 2^{12} x 15 = 61,440 bits of ROM. The diode array represents a factor of 23 reduction in required memory. This is a rather astounding difference which demonstrates that the real power of this array is its ability to pick unique and random sets of minterms from large lists of input data.

A second example is the control section of a CO_2 Laser Scribe system used to scribe the sapphire wafers. This system controls an XY stage for positioning the wafer and controls firing of the laser during scribing operations. A block diagram of the system is shown in Figure 6. The diode array outputs 4 bit words for both X and Y coordinates, sequential network set/reset commands, and the laser enable. A preprogrammed sequence of scribe paths is stored in the array. At the end of each scribe pass a new set of coordinates is read out from the array. The program is modified by a wafer size switch which selects the appropriate number of paths in both directions to ensure complete scribing of the wafer and a mode control which selects full automatic or single path operation. The array has 15 outputs which are functions of 11 input variables and 2 enables. This is equivalent to ROM storage of 2048 8-bit words plus 16 TTL gates with 74 inputs (approximately 10 fourteen-pin packages). The section of the array dedicated to the listing of table coordinates is 115 rows x 40 columns = 4600 diodes. The ROM capacity needed for this is 16,384 bits (2048 x 8), almost four times greater than used in the diode array without considering the logic pattern. Here the logic is "free" in the sense that all required inputs are already connected to the array and all that is needed to eliminate 10 packages is 7 output sensors (two 14 pin packages), plus the encoding of 12 more rows of the array.

In each of these cases, as in all logic and memory applications of the diode array, a single delay from input to output (typically 20 nsec) even for the most complex logic functions maximizes system cycle rates. If four levels of standard TTL gate logic were required to mechanize some function, a typical delay through the gates would be 4 x 11 nsec (Tpdl) = 44 nsec. The diode array is twice as fast.

Economics of Diode Arrays

The examples given have shown several ways in which the diode array can reduce system size and cost, and improve performance. The economics of diode arrays depends on how much cost can be cut by using them. A recent article(6) has delved into cost considerations of putting systems together. It discusses the effect reduced part count has on system economics. Four main points are: 1. Reduced spare parts inventory, 2. Reduced design, layout, and debug effort, 3. Lower manufacturing cost, 4. Meantime to repair and service training costs are reduced.

The price of inserting one IC in a system must include labor, connectors, PC boards, cabling, etc. If the IC were free, a major portion of the cost would still remain. This forces designers toward more complex packagable functions (MSI and LSI) to reduce the number of components.

If ROM's are used to replace logic, the system cost is a function of both cost/bit of the ROM and the effective number of bits required to replace a gate. The referenced article indicates that 12 bits/gate is a realistic number.

Several programmable ROM's are on the market which allow the designer to encode his information relatively quickly at his own lab. However, in the two system examples given, if ROM replacement of logic were used, a much larger memory than the diode array would be needed because of the inability of standard ROM's to take advantage of minimization and "don't care" conditions of the input data.

Another advantage of the diode array is that system design changes can be implemented quickly and inexpensively. If changes in memory or logic equations are required, a new diode array can be quickly encoded to replace the part with an error. The laser encoding process gives fast turnaround to single quantity items with no programming charge.

CONCLUSIONS

Using a large scale integrated diode array in system design does optimize both system and design by improving system performance (increased cycle rates, better data organization), reducing design and debug effort, (standardized parts and interfaces) reducing parts count (a single part type can be multifunctional, plus system size and power requirements also decrease), and lowering the overall system cost. The flexibility of organization of stored data possible with the array provides a means for optimizing systems and system designs.

ACKNOWLEDGEMENTS

I wish to thank Dr. James A. Luisi and Mr. Quentin C. Cassen for their abundent help and encouragement.

BIBLIOGRAPHY

1. Fletcher, William J., and Despain, Alvin M., "Simplify Sequential Circuit Designs," Electronic Design, June 24, 1971, pp 72-73.

2. Read-Only Memory Applications of the Silicon-on-Sapphire 40 x 128 Diode Array, Application Note No. 1, North American Rockwell Microelectronics Company, June 1971.

3. Sypherd, Allen D., "Design of Digital Filters Using Read-Only Memories," proceeding of NEC, Volume 25, 1969, pp 691-693.

4. Elliott, D. F., and Sypherd, Allen D., "Simplify Function Generator Design," Electronic Design, October 11, 1970, pp 86-89.

5. Sypherd, Allen D., and Salman, Naif D., "Automatic Laser Encoding of Semiconductor Read-Only Memories," Proc. of NEC, Volume 24, 1968, pp. 206-209.

6. Davidow, William H., "General Purpose Microcontrollers Part 1: Economic Considerations," Computer Design, July 1972, pp. 75-79.

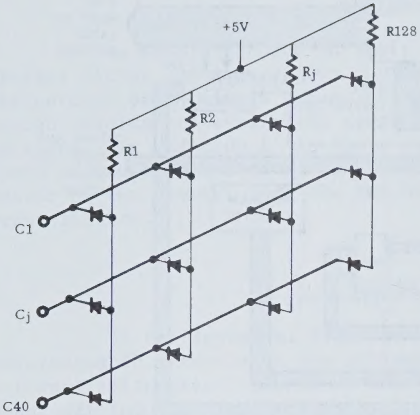
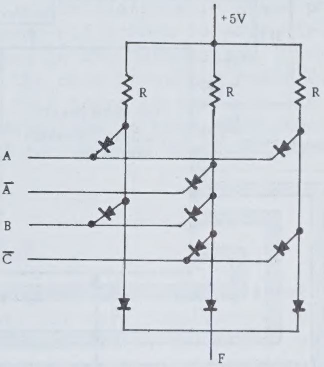

+5V
R128
Rj
R2
R1
C1
Cj
C40

Figure 1. Schematic of 40 x 128
SOS Diode Array


+5V
R
R
R
A
Ā
B
C̄
F

$F = AB + \bar{A}\bar{C}B + A\bar{C}$

Figure 2. Diode Array Logic Mechanization
of F

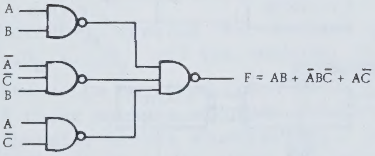
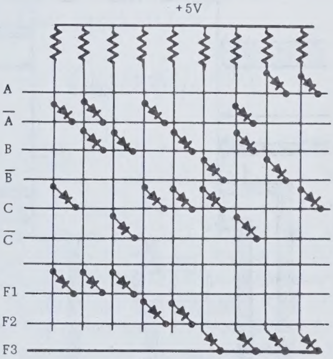

A
B
Ā
C̄
B
A
C̄
F = AB + ĀBC̄ + AC̄

Figure 3. Logic Gate Mechanization of F


+5V
A
Ā
B
B̄
C
C̄
F1
F2
F3

$F1 = \bar{A}C + \bar{A}B + B\bar{C}$

$F2 = \bar{A}C + BC$

$F3 = \bar{B}C + \bar{A}B\bar{C} + A\bar{B} + AC$

Figure 4. Diode Array Mechanization of
Optimized Truth Table Eq. 5

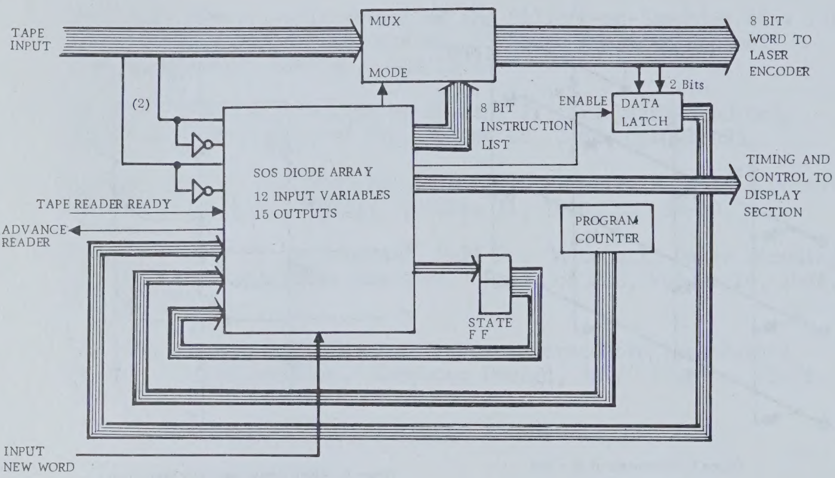

TAPE INPUT
MUX
MODE
8 BIT WORD TO LASER ENCODER
2 Bits
(2)
8 BIT INSTRUCTION LIST
ENABLE
DATA LATCH
SOS DIODE ARRAY
12 INPUT VARIABLES
15 OUTPUTS
TIMING AND CONTROL TO DISPLAY SECTION
TAPE READER READY
ADVANCE READER
PROGRAM COUNTER
STATE F F
INPUT NEW WORD

Figure 5. Block Diagram of Laser Encode Test Subsystem using the Diode Array

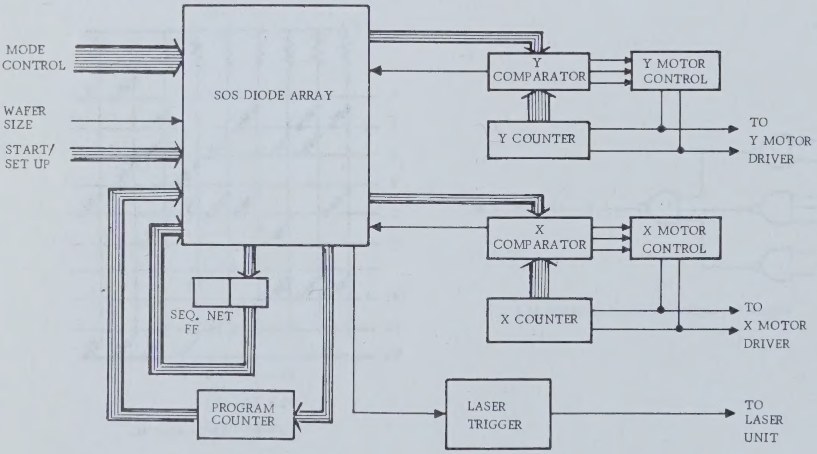

MODE CONTROL
WAFER SIZE
START/ SET UP
SOS DIODE ARRAY
Y COMPARATOR
Y MOTOR CONTROL
Y COUNTER
TO Y MOTOR DRIVER
X COMPARATOR
X MOTOR CONTROL
X COUNTER
TO X MOTOR DRIVER
SEQ. NET FF
PROGRAM COUNTER
LASER TRIGGER
TO LASER UNIT

Figure 6. Block Diagram of CO_2 Laser Scriber Control System using SOS Diode Array

AN ABSOLUTE BOUND ON LIMIT CYCLES DUE TO ROUND-OFF ERRORS IN DIGITAL FILTERS†

JAMES L. LONG
TIMOTHY N. TRICK
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

An absolute bound on limit cycle oscillations in fixed point digital filter implementations due to round-off errors is presented. Periodicity of the limit cycles is assumed in the derivation. Useful design results are explicitly given for the case of second order filter sections. In addition it is shown that this bound is equal to the RMS bound of Sandberg and Kaiser for real roots, and is never more than a factor of two greater than the RMS bound for complex roots for second order filters.

I. INTRODUCTION

In the design of fixed point digital filters one is particularly interested in determining the minimum bit per word requirement for the internal arithmetic. Bounds on the maximum value of the error due to round-off induced limit cycles are of prime importance in making that determination. In this paper a new bound on the maximum amplitude of limit cycles in fixed point digital filters is presented.

II. DERIVATION OF AN ABSOLUTE BOUND FOR PERIODIC LIMIT CYCLES

Consider the Nth-order nonlinear difference equation (1) representing the direct form digital filter realization

$$y(n) = -[\alpha_1 y(n-1)]_r - [\alpha_2 y(n-2)]_r - \cdots -[\alpha_N y(n-N)]_r + x(n) \qquad (1)$$

where $[\cdot]_r$ denotes the standard quantization function. The coefficients, $\alpha_1, \alpha_2, \ldots, \alpha_N$, and the numbers, $y(-N), y(-N+1), \ldots, y(-1), y(0), \ldots$, are assumed to be fixed point machine numbers. Round-off after multiplication of these numbers accounts for the need of quantization functions in the otherwise linear equation (1). Each product is rounded before addition takes place as is often the case in practical filters to alleviate the need for double precision registers. Throughout this paper the independent forcing term $x(n)$ will be taken to be zero and the coefficients in (1) are such that the equation without quantizers is asymptotically stable.

[†] This work was supported by the Joint Services Electronics Program (U. S. Army, U. S. Navy, U. S. Air Force) under Contract DAAB-07-67-C-0199.

Equation (1) may be replaced by a linear difference equation driven by round-off induced error sequences, $e_1(n), e_2(n), \ldots, e_N(n)$, as follows:

$$y(n) = -\alpha_1 y(n-1) + e_1(n) - \alpha_2 y(n-2) + e_2(n) - \cdots - \alpha_N y(n-N) + e_N(n). \tag{2}$$

Assuming a normalized quantization width of 1, the error terms in (2) are bounded in magnitude by 1/2. A single error term may be defined as follows:

$$e(n) \overset{\Delta}{=} e_1(n) + e_2(n) + \cdots + e_N(n) \tag{3}$$

where $|e(n)| \leq \frac{N}{2}$.

Now suppose that the limit cycle is periodic of period M, then the error input sequence has the property that

$$e(k) = e(k+M) \tag{4}$$

for all k. The steady state output due to the round-off induced error, e(n), is given by the discrete convolution

$$y(n) = \sum_{k=-\infty}^{n} h(n-k)e(k) \tag{5}$$

which can be decomposed in a manner directly analogous to the method of Trick and Anderson [3] for the case of a continuous convolution. First, (5) is expressed as a double summation

$$y(n) = \sum_{i=0}^{\infty} \Big[\sum_{k=n-(i+1)M+1}^{k=n-iM} h(n-k)e(k) \Big]. \tag{6}$$

Letting $p \overset{\Delta}{=} n-iM-k$ one obtains

$$y(n) = \sum_{i=0}^{\infty} \Big[\sum_{p=0}^{p=M-1} h(p+iM)e(n-iM-p) \Big]. \tag{7}$$

Interchanging the summation in (7) and noting that $e(n-p-iM) = e(n-p)$ we have

$$y(n) = \sum_{p=0}^{M-1} e(n-p) \Big[\sum_{i=0}^{\infty} h(p+iM) \Big]. \tag{8}$$

The desired bound follows directly from (8) as

$$|y(n)|_{\max} = \frac{N}{2} \sum_{p=0}^{M-1} \left| \sum_{i=0}^{\infty} h(p+iM) \right| \tag{9}$$

for any value of n.

A bound on the right hand side of (9) for any M and the asymptotic behavior for large M are of particular interest. Considering (9) it is obvious that

$$\sum_{p=0}^{M-1} \left| \sum_{i=0}^{\infty} h(p+iM) \right| \leq \sum_{p=0}^{M-1} \sum_{i=0}^{\infty} |h(p+iM)| . \tag{10}$$

Letting $k \overset{\Delta}{=} p+iM$ and interchanging order of summation in (10) gives

$$\sum_{p=0}^{M-1} \sum_{i=0}^{\infty} |h(p+iM)| = \sum_{i=0}^{\infty} \sum_{k=iM}^{k=(i+1)M-1} |h(k)| = \sum_{k=0}^{\infty} |h(k)| . \tag{11}$$

Combining (9) and (11) one gets

$$|y(n)|_{\max} \leq \frac{N}{2} \sum_{k=0}^{\infty} |h(k)| \tag{12}$$

which is independent of the value of n and M. Utilizing the asymptotic stability property assumed on h(p) one can show that

$$\lim_{M\to\infty} \sum_{p=0}^{M-1} \left| \sum_{i=0}^{\infty} h(p+iM) \right| = \sum_{p=0}^{\infty} |h(p)| . \tag{13}$$

III. ABSOLUTE BOUND FOR SECOND ORDER FILTERS

Equation (9) can be reduced to an explicit function of the filter coefficients for the commonly used second order filter. Consider the second order transfer function

$$H(z) = \frac{z^2}{z^2 + az + b} , \tag{14}$$

where letters a and b are used in place of α_1 and α_2 to agree with standard notation. The inversion formula for (14) is

$$h(p) = \frac{1}{2\pi j} \int_c \frac{z^2}{z^2 + az + b} \cdot z^{p-1} dz \tag{15}$$

where c is taken to be the unit circle, therefore

$$\sum_{i=0}^{\infty} h(p+iM) = \sum_{i=0}^{\infty} \{\frac{1}{2\pi j} \int_{c} \frac{z^{p+iM+1}}{z^{2}+az+b}\, dz\}. \tag{16}$$

Combining the results obtained by evaluating the right hand side of (16) by the residue theorem with equation (9), one gets for the case of distinct roots of the equation z^2+az+b

$$|y(n)|_{max} = \sum_{p=0}^{M-1} \left| \frac{1}{2\sqrt{(\frac{a}{2})^{2}-b}} \left[\frac{(-\frac{a}{2}+\sqrt{(\frac{a}{2})^{2}-b})^{p+1}}{1-(-\frac{a}{2}+\sqrt{(\frac{a}{2})^{2}-b})^{M}} - \frac{(-\frac{a}{2}-\sqrt{(\frac{a}{2})^{2}-b})^{p+1}}{1-(-\frac{a}{2}-\sqrt{(\frac{a}{2})^{2}-b})^{M}} \right] \right| \tag{17}$$

and for the repeated root case

$$|y(n)|_{max} = \sum_{p=0}^{M-1} \left| (-\frac{a}{2})^{p} \left[\frac{p}{[1-(-\frac{a}{2})^{M}]} + \frac{1+(-\frac{a}{2})^{M}(M-1)}{[1-(-\frac{a}{2})^{M}]^{2}} \right] \right| . \tag{18}$$

Note that (17) and (18) are computationally more attractive than the "matrix" method of Parker and Hess [2].

Two important special cases of the "periodic" bound (17) are when M=1 and M=2, corresponding to frequencies of dc and one-half the sampling frequency:

$$|y(n)|_{max} = \frac{1}{1+a+b} \quad , \quad M=1 \tag{19}$$

$$|y(n)|_{max} = \frac{1}{1+b-|a|} \quad , \quad M=2 \tag{20}$$

These bounds are the same as those derived by Jackson [5] for each of the above special cases, and they are also identical to the results obtained by Parker and Hess [2] with their matrix bound and by Kaiser and Sandberg [1] with their RMS bound.

In general, the period of the limit cycle is not known, which makes the summation of the absolute value of the impulse response an attractive bound since it is the least upper bound on (17) independent of M. The evaluation of $\sum_{n=0}^{\infty} |h(n)|$ can be done exactly for the cases of real roots by straightforward summation of terms yielding the following:

$$\sum_{n=0}^{\infty} |h(n)| = \frac{1}{1 - |a| + b} \tag{21}$$

for distinct real roots, and

$$\sum_{n=0}^{\infty} |h(n)| = \frac{1}{(1 - \sqrt{b})^2} \tag{22}$$

for a double real root where $a^2 = 4b$.

Obtaining an exact expression for the case of complex roots is difficult, since the impulse response of a second order filter with complex roots has the form

$$h(n) = \frac{r^n}{\sin \theta} \sin(n+1)\theta \tag{23}$$

where $r = \sqrt{b}$, $\theta = \cos^{-1}(\frac{-a}{2\sqrt{b}})$, and $4b > a^2$. In this case the least upper bound on the absolute value of the round-off error is

$$\sum_{n=0}^{\infty} |h(n)| = \sum_{n=0}^{\infty} \left|\frac{r^n}{\sin \theta} \sin(n+1)\theta\right| . \tag{24}$$

The following two estimates of (24) can be useful for design purposes:

$$\sum_{n=0}^{\infty} |h(n)| \leq \sum_{n=0}^{\infty} \left|\frac{r^n}{\sin \theta}\right| = \frac{1 + \sqrt{b}}{(1-b)\sqrt{1 - \frac{a^2}{4b}}} \tag{25}$$

or

$$\sum_{n=0}^{\infty} |h(n)| \leq \sum_{n=0}^{\infty} (n+1)r^n = \frac{1}{(1 - \sqrt{b})^2} . \tag{26}$$

Thus, the absolute value of the round-off error is bounded by

$$|y(n)|_{max} \leq \begin{cases} \dfrac{1}{1 - |a| + b} , & \text{for either } b \leq 0 \text{ or } b > 0 \text{ and } 2\sqrt{b} \leq |a| \quad (27) \\[2ex] \dfrac{1}{(1 - \sqrt{b})^2} , & \text{for } b > 0 \text{ and } 2b\sqrt{\dfrac{2}{\sqrt{b}} - 1} \leq |a| \leq 2\sqrt{b} \quad (28) \\[2ex] \dfrac{(1 + \sqrt{b})}{(1-b)\sqrt{1 - \dfrac{a^2}{4b}}} , & \text{for } b > 0 \text{ and } |a| \leq 2b\sqrt{\dfrac{2}{\sqrt{b}} - 1} \quad (29) \end{cases}$$

IV. COMPARISON OF RESULTS

A computer program was used to evaluate the bound (17) for the case M=7 and b=.9 while the parameter a was stepped from -1.7 to 1.7. The results of this computation were identical to those displayed graphically for the same parameter values by Hess [4] using his "matrix" method. Figures 1 and 2 were also constructed using equation (17). Fixed values of a and b were used while M was varied. The RMS bound of Kaiser and Sandberg [1] as well as the absolute bound derived in this paper are displayed on the graphs. One can see by the asymptotic behavior for large M that the absolute bound for complex roots is not the least upper bound due to the approximations made in (25) and (26). Values of the bounds in Figures 1 and 2 can also be compared to the heuristic bound

$$|y(n)|_{\max} \le \text{integer part of } [\frac{.5}{1-b}] \tag{30}$$

which was derived by Jackson [5] using an "effective-value model." This bound yields numerical values of 5 for the example illustrated in Figure 1 and 50 for the example illustrated in Figure 2. However, Parker and Hess [2] have shown by example that (30) does not always bound the limit cycle, whereas (27)-(29) does.

Interesting comparisons can also be made between the absolute bounds (27), (28), (29) and the RMS bounds obtained by Sandberg and Kaiser. Their results are as follows (using notation consistent with this paper):

$$(\frac{1}{M}\sum_{n=0}^{M-1} y^2(n))^{1/2} \le \rho(a,b) \tag{31}$$

for any M > 0 and where

$$\rho(a,b) = \begin{cases} \dfrac{1}{1+b-|a|}, & \text{for either } b \le 0 \text{ or } b>0 \text{ and } |a| \ge \dfrac{4b}{1+b} \quad (32) \\[2ex] \dfrac{1}{(1-b)(1-\frac{a^2}{4b})^{1/2}}, & \text{for } b>0 \text{ and } |a| < \dfrac{4b}{1+b} \quad (33) \end{cases}$$

One should make the important observation that (32) is identical to (27) over the region where $a^2-4b \ge 0$. Therefore, the RMS bound (32) is also an absolute bound over the region $a^2-4b \ge 0$. Comparison of (29) and (33) over the region $b > 0$ and $|a| < \frac{4b}{1+b}$ shows that they differ by a factor of $(1+\sqrt{b})$. Note that (29) cannot exceed (33) by more than a factor of two over the specified region and for small values of b they are approximately the same. The following important result can be shown by careful comparison of the bounds:

$$\frac{|y(n)|_{\max}}{\rho(a,b)} \le 2 \tag{34}$$

for any a and b for which the linear filter is asymptotically stable. Realizing this fact, the designer using the RMS bound can be certain that the maximum value of the limit cycle will not exceed this bound by a factor of more than two.

REFERENCES

1. Sandberg, I. W. and Kaiser, J. F., "A Bound on Limit Cycles in Fixed-Point Implementations of Digital Filters," IEEE Trans. on Audio and Electroacoustics, Vol. AU-20, pp. 110-112, June 1972.

2. Parker, S. P. and Hess, S. F., "Limit-Cycle Oscillations in Digital Filters," IEEE Trans. on Circuit Theory, Vol. CT-18, No. 6, pp. 687-697, Nov. 1971.

3. Trick, T. N. and Anderson, D. R., "Stability of Nonlinear Networks," IEEE Trans. on Circuit Theory, Vol. CT-16, No. 3, pp. 302-311, Aug. 1969.

4. Hess, S. F., "A Deterministic Analysis of Limit Cycle Oscillations in Recursive Digital Filters Due to Quantization," Ph.D. Dissertation, Naval Postgraduate School, Monterey, Calif., Dec. 1970.

5. Jackson, L. B., "An Analysis of Limit Cycles Due to Multiplication Rounding in Recursive Digital (Sub) Filters," Proc. 7th Annual Allerton Conf. on Circuit and System Theory, pp. 69-78, Oct. 1969.

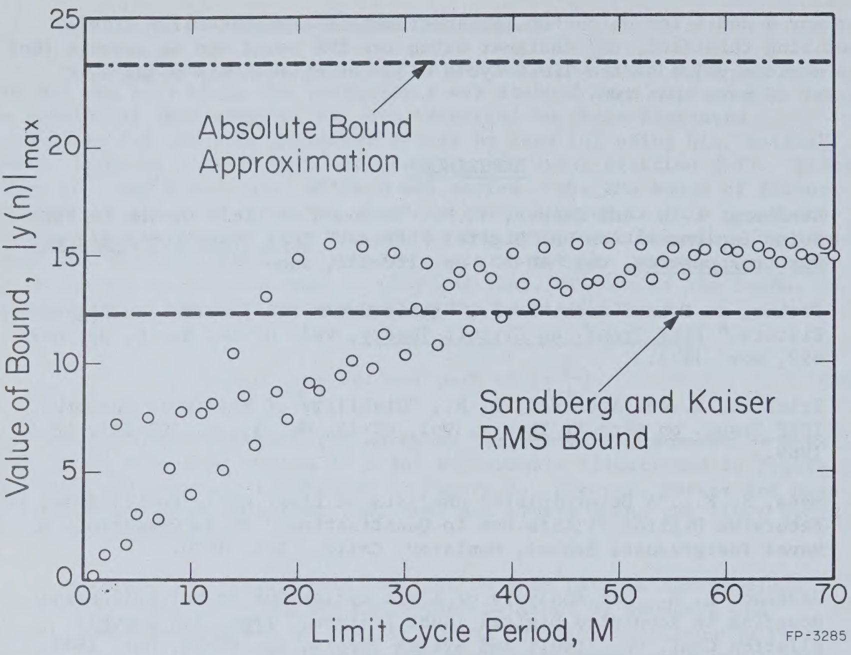


Figure 1. Amplitude bound for a = 1.09 and b = 0.9.

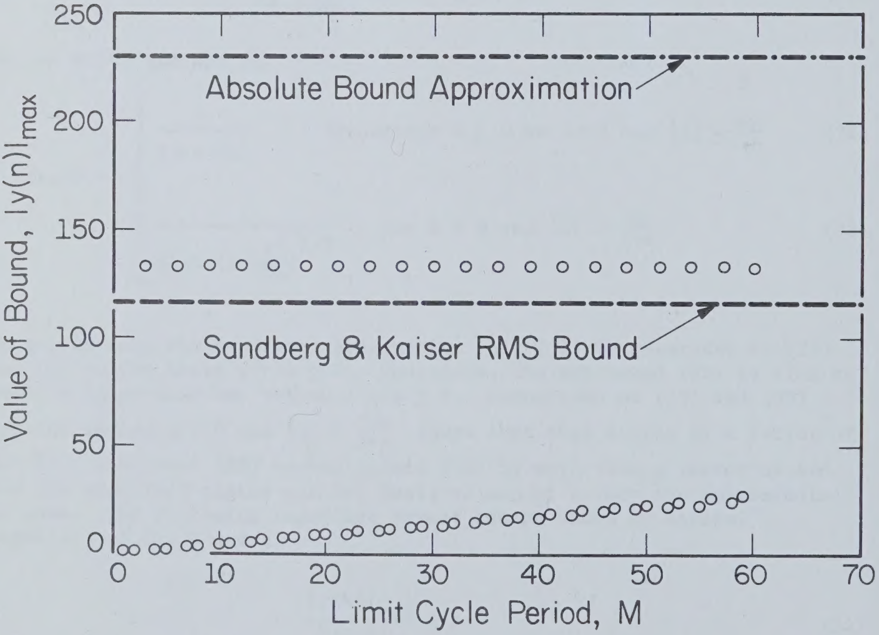


Figure 2. Amplitude bound for a = 1.0 and b = 0.99.

DISTRIBUTED R-C TIMING CIRCUITS IN MULTIVIBRATORS

R.A. HAYES   J.J. BOURQUIN
Engineering Research Laboratory
University of Missouri-Rolla
Rolla, Missouri 65401

ABSTRACT

This paper analyzes two monostable multivibrators using uniformly distributed networks. One circuit utilizes the standard distributed device consisting of conductor, dielectric, and resistor layers. The other circuit utilizes a resistor-dielectric-resistor layered device. Pulse duration equations are derived for both circuits and compared to a discrete version.

INTRODUCTION

Much of the work that has been done to date in the distributed networks area has been in the frequency domain. Work in the time domain has mainly been concerned with impulse and step responses of distributed networks. A few papers have been published concerning work in the time domain[1], but few papers have been published concerning electronic circuits using distributed network devices[2].

This paper considers the use of distributed network devices in monostable multivibrators. A basic monostable multivibrator circuit using discrete components is shown in fig. (1). Note that trigger pulse shaping elements and speed-up capacitors have been deleted for purposes of simplification. This circuit has been analyzed in detail in the literature[3,4]. After the appropriate assumptions of $V_{CE_{SAT}} = 0$ and $V_{CC} >> V_{BE}$, the approximate pulse duration is

$$t = \text{pulse duration} = 0.69\ R.C. \qquad (1)$$

where R and C are defined in fig. (1).

MONOSTABLE MULTIVIBRATOR USING URC

The first circuit using a distributed network is shown in fig. (2). Note that the capacitor C and resistor R_1 in fig. (1) have been replaced by a single URC element in fig. (2) with all other elements remaining the same.

Analysis of the circuit begins by considering the initial voltage distribution on the URC before a trigger pulse is applied. As shown in fig. (3a) T_1 is initially cut off and T_2 is saturated, resulting in a voltage distribution of $v(x) = (V_{BE_2} - V_{CC})\frac{x}{d}$. After the application of a trigger pulse, T_1 switches from a cutoff state to a saturation state bringing the voltage at the conductive terminal of the URC to $V_{CE_{SAT_1}}$. This switching effect couples the negative initial voltage v(d) to the base of T_2 driving it into cut off and thus bringing V_O from $V_{CE_{SAT_2}}$ to V_{CC}. This switching action establishes boundary and initial conditions on the URC to be used with the partial differential equation governing the URC as shown in fig. (3b). Fig. (3c) references ground as the conductive layer of the URC for easy solution of the timing circuit.

The pulse duration equation is established by generating the solution for V_{BE_2} (fig. 3c) and recognizing that the output, V_o, will return to the stable state when the voltage, V_{BE_2}, is great enough to turn on T_2. The solution, V_{BE_2}, of the circuit in fig. (3c) is found by solving the RC transmission line equation with the appropriate initial and boundary conditions.

$$\frac{\partial^2 v(x,s)}{\partial x^2} - Z(s)\ Y(s)\ V(x,s) = -Z(s) c v(x,0) \qquad (2)$$

where v(x,s) is the Laplace Transform with respect to t of the voltage on the line.

$Z(s) = R$ = total resistance of the line
$Y(s) = C$ = total capacitance of the line
$v(x,0) = (V_{BE} - V_{CC})(\frac{x}{d})$ = initial voltage distribution

The total solution for V(d,s) is

$$V(d,s) = A(s)e^{\sqrt{sRC}} + B(s)e^{-\sqrt{sRC}} + \frac{K_1}{s}$$

where $K_1 = V_{BE} - V_{CC}$
and A(s) and B(s) are constants to be determined by the boundary conditions

$$V(0,s) = V_{CC} - V_{CE_{SAT_1}} = K_2$$

$$I(d,s) = 0$$

Evaluating the constant A(s) and B(s) and combining terms yields

$$V(d,s) = \frac{K_2}{s}\operatorname{sech}\theta - \frac{K_1}{s\theta}\tanh\theta + \frac{K_1}{s} \qquad (3)$$

where $\theta = \sqrt{sRC}$

$$\mathcal{L}^{-1}\left[\frac{\operatorname{sech}\sqrt{Ts}}{s}\right] = 1 + \frac{4}{\pi}\sum_{n=1}^{\infty}(-1)^n e^{-\frac{(2n-1)^2\pi^2 t}{4T}} \qquad (4)$$

$$\mathcal{L}^{-1}\left[\frac{1}{s}\ \frac{\tanh\sqrt{Ts}}{\sqrt{Ts}}\right] = 1 - \frac{8}{\pi^2}\sum_{n=1,3,5,\cdots}^{\infty}\frac{1}{n^2}e^{-\frac{n^2\pi^2 t}{4T}}$$

as found by using a theorem from the Bateman Manuscript Project [5,P.132].

Taking the inverse Laplace Transform of eq. (3) yields

$$v(d,t) = K_2 + \sum_{n=1,3,5,\cdots}^{\infty} e^{-\frac{n^2\pi^2 t}{4T}} \left[\frac{4K_2}{\pi} \frac{(-1)^{\frac{n+1}{2}}}{n} + \frac{K_1 8}{\pi^2} \frac{1}{n^2} \right]$$

The actual voltage expression for the base voltage is $V_{B_2}(t) = V(t) + V_{CE_{SAT}}$

$$= V_{CC} + \sum_{n=1,3,5,\cdots}^{\infty} e^{-\frac{n^2\pi^2 t}{4T}} \left[\frac{4(V_{CC}-V_{CE_{SAT}})(-1)^{\frac{n+1}{2}}}{\pi n} + \frac{(V_{BE}-V_{CC})(8)}{\pi^2 n^2} \right] \quad (5)$$

Eq. (5) can be further simplified by making the following approximation:

$$V_{CE_{SAT}} \approx 0$$

$$V_{BE} \approx 0$$

which reduces the equation to the following

$$-1 = \sum_{n=1,3,5,\cdots}^{\infty} e^{-\frac{n^2\pi^2 t}{4T}} \left[\frac{4}{\pi n}(-1)^{\frac{n+1}{2}} - \frac{8}{\pi^2 n^2} \right]$$

Truncating the series after the first term results in the following expression:

$$t \approx 0.3RC = 0.3T \quad (6)$$

which is the approximate pulse duration for the circuit. Actual circuit data and computer simulations of the circuit indicate the pulse duration to be t = 0.31T. Thus the truncation of the infinite series and approximation of $V_{BE} = V_{CE_{SAT}} = 0$ have produced a 3.3% error in the pulse duration. Computer simulations were done using a 20 section R-C ladder line on ECAP. Fig. (4) shows an actual trace from an URC monostable multivibrator. The vertical scale is 5v/cm and horizontal scale is 50μsec/cm with R = 27.8K and C = 0.028μf.

MONOSTABLE MULTIVIBRATOR USING DOUBLE RESISTIVE URC

The monostable multivibrator using an RCR distributed device is shown in fig. (5). The RCR circuit has replaced the collector and bias resistors and coupling capacitor shown in fig. (1) by a single multilayered distributed device. Similar techniques are utilized in developing the pulse duration equation as were employed for the first monostable multibivrator.

Initial conditions are established by considering the circuit in its stable state. T_1 is cut off and T_2 is saturated as shown in fig. (6a) resulting in the following voltage distribution on the network: $v(x) = -(V_{CC} - V_{BE})(\frac{x}{d})$. Application of a triggering pulse drives T_1 into saturation causing T_2 to switch from saturation to cutoff. This switching action holds T_2 in a cutoff state until the voltage reaches sufficient amplitude to turn on T_2 causing the circuit to return to its stable state. The voltage solution, $V_{B_2}(t)$, of the RCR distributed network will be generated with the appropriate initial and boundary conditions (fig. 6b). Two boundary conditions necessary for the solution of this circuit are $I_1(d,s) = 0$ and $V(0,s) = 0$. The other boundary condition, $I_2(d,s) = \frac{V_{CC}}{R_2 s}$, is an approximation, however a computer program was written[2] which verifies the validity of such an approximation.

Partial differential equations governing the RCR network have been derived[6] and consist of the following:

$$\frac{\partial I_1(x,t)}{\partial x} = -C\frac{\partial v(x,t)}{\partial t} \tag{7}$$

$$\frac{\partial I_2(x,t)}{\partial x} = C\frac{\partial v(x,t)}{\partial t} \tag{8}$$

$$\frac{\partial v(x,t)}{\partial x} = R_1 I_1(x,t) + R_2 I_2(x,t) \tag{9}$$

Differentiating eq. (9) and substituting eqs. (7) and (8) yields

$$\frac{\partial^2 v(x,t)}{\partial x^2} = \frac{\partial v(x,t)}{\partial t} C(R_1 + R_2)$$

Letting $\tau = C(R_1 + R_2)$ and taking the Laplace Transform with respect to t yields

$$\frac{d^2 v(x,s)}{dx^2} = \tau s V(x,s) - \tau v(x,0) \tag{10}$$

The homogeneous solution to eq. (10) is $V(x,s) = A\,e^{\gamma x} + B\,e^{-\gamma x}$ where $\gamma = \sqrt{\tau s}$

A particular solution is $-(V_{CC} - V_{BE})(\frac{x}{d})$ producing a total solution

$$V(x,s) = Ae^{\gamma x} + Be^{-\gamma x} - (V_{CC} - V_{BE})(\frac{x}{d}) \tag{11}$$

Invoking the voltage boundary condition at x = 0

$v(0,s) = 0.0$ yields $A = -B$.

Evaluation of A requires the use of eq. (9) and the current boundary conditions $I_1(d,s) = 0$, $I_2(d,s) = \frac{V_{CC}}{R_2 s}$, and yields

$$A = \frac{(2V_{CC} - V_{BE})/s}{2\gamma \cosh\gamma}$$

Substituting the expressions for A and B and evaluating eq. for x = d yields

$$V(d,s) = \frac{(2V_{CC} - V_{BE})}{s\gamma} \tanh(\gamma d) - \frac{(V_{CC} - V_{BE})}{s} \qquad (12)$$

$$V_{B_2} = V(d,s) + V_{CE_{SAT}}/s$$

applying the inverse Laplace Transform of eq. (4) to eq. (12) yields

$$v(d,t) = (2V_{CC}-V_{BE})\left[1-\frac{8}{\pi^2}\sum_{n=1,3,5,\cdots}^{\infty}\frac{1}{n^2}\, e^{-\frac{n^2\pi^2 t}{4\tau}}\right] - (V_{CC}-V_{BE})u(t) + V_{CE_{SAT}}u(t) \qquad (13)$$

This eq. can be further simplified by letting $V_{BE} = V_{CE_{SAT}} = 0$, yielding

$$\frac{\pi^2}{16} = \sum_{n=1,3,5,\cdots}^{\infty}\frac{1}{n^2}\, e^{-\frac{n^2\pi^2 t}{4\tau}}$$

Truncation of the infinite series after the first term and solving for t results in an approximation for the pulse duration

$$t = 0.197\tau. \qquad (14)$$

A computer program has been written which validates the approximations of $V_{BE} = V_{CE_{SAT}} = 0$ and truncation of the infinite series. Also an ECAP analysis of the circuit using a 20 section approximation to the RCR distributed network has been performed resulting in a 6.5% error between actual and derived pulse duration expressions.

CONCLUSION

As can be seen from figs. (1), (2), and (5), distributed networks have been used to reduce the total number of passive elements required in a monostable multivibrator. This reduction becomes more effective when these same distributed elements are implemented in astable multivibrators. As can be seen from comparison of eqs. (1), (6), and (14), monostable multivibrators using distributed networks require larger values of capacitance than the discrete version.

An astable multivibrator using RCR networks is shown in fig. (7). The period of the astable output can be derived as a simple extension from the monostable model. The period for the astable multivibrator is $T = 0.197[C_1(R_1+R_2) + C_2(R_3+R_4)]$ providing the transistors are in saturation when the on state and recover completely when cut off.

REFERENCES

1. Chang, F. Y., "Time-Domain Synthesis Using Multilayer RC Distributed Networks," IEEE Transactions on Circuit Theory, Vol. CT-17, No. 4., Nov. 1970.
2. Hager, Charles K., "Distributed Parameter Networks for Circuit Miniaturization," Proceedings of Electronics Components Conference, 1960.
3. Strauss, Leonard, *Wave Generation and Shaping*, McGraw-Hill Book Company, New York, 1970.
4. Millman, Jacob, and Herbert Taub, *Pulse, Digital, and Switching Waveforms*, McGraw-Hill Book Company, New York, 1965.
5. Bateman Manuscript Project, *Tables of Integral Transforms*, Vol. I, McGraw-Hill Book Company.
6. Chirlian, Paul M., *Integrated and Active Network Analysis and Synthesis*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1967.

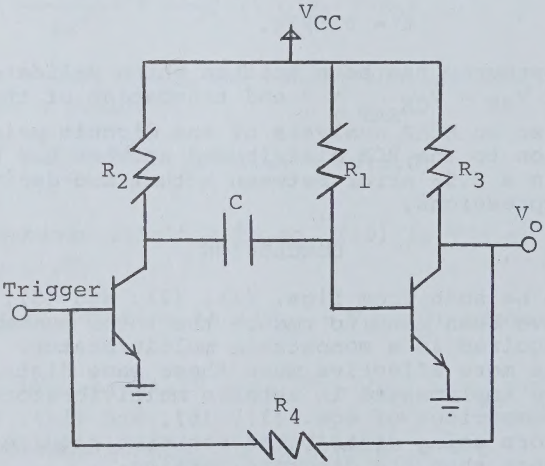


Fig. (1) Discrete Monostable Multivibrator

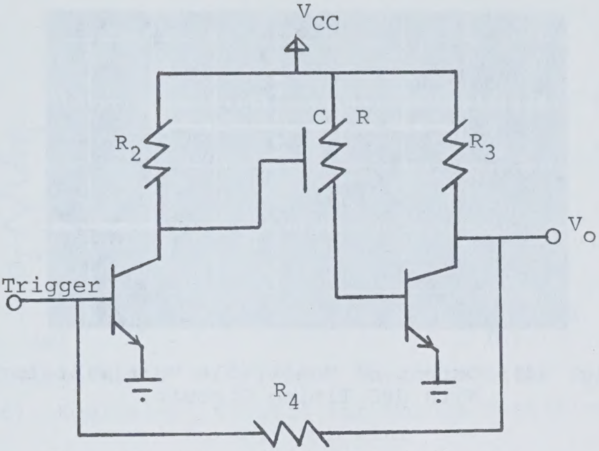
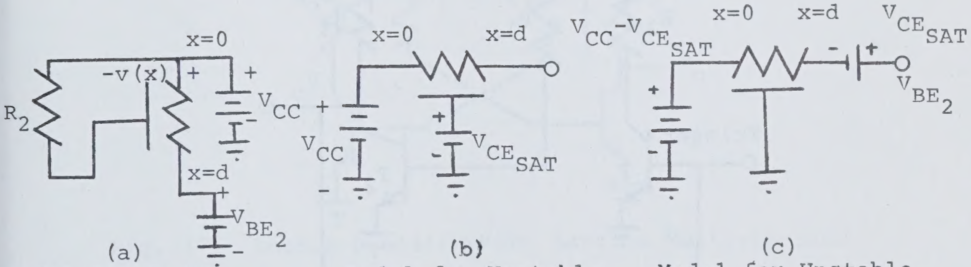

V_{CC}
C
R
R_2
R_3
V_o
Trigger
R_4

Fig. (2) Monostable Multivibrator
Using URC


x=0
-v(x)
+
+
R_2
V_{CC}
-
x=d
+
V_{BE_2}
-
x=0
x=d
+
V_{CC}
-
+
-
$V_{CE_{SAT}}$
$V_{CC}-V_{CE_{SAT}}$
x=0
x=d
$V_{CE_{SAT}}$
+
-
-
+
V_{BE_2}
(a)
(b)
(c)

Model for Stable State

Model for Unstable State

Model for Unstable State

Fig. (3) Equivalent Circuits for Monostable Multivibrator

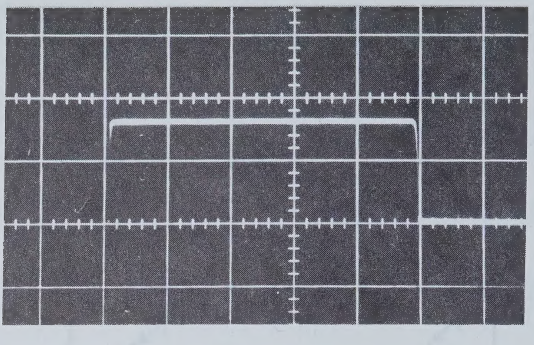


Fig. (4) Output of Monostable Multivibrator With URC Timing Circuit

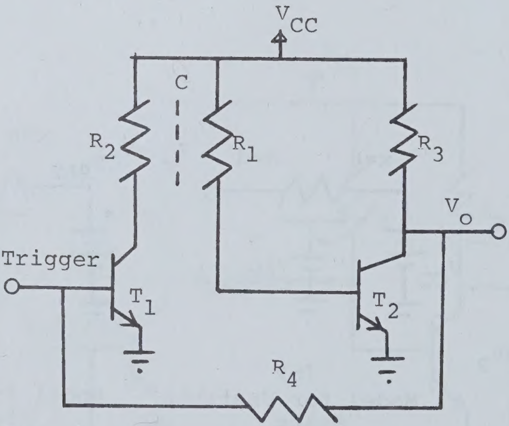
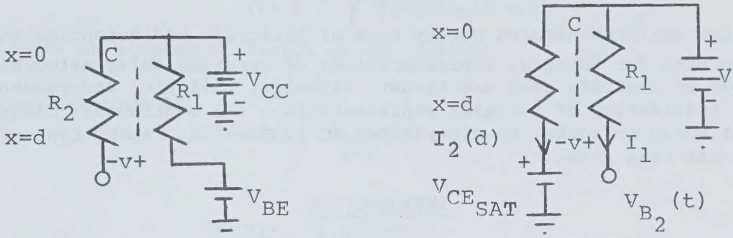

V_{CC}
C
R_2
R_1
R_3
V_o
Trigger
T_1
T_2
R_4

Fig. (5) Monstable Multivibrator Using Double-resistive URC.


x=0
C
R2
R1
+
VCC
-
x=d
-v+
VBE
(a)
x=0
C
R1
+
VCC
-
x=d
I2(d)
-v+
I1
+
VCE SAT
VB2 (t)
(b)

Stable State Model   Unstable State Model

Fig. (6) Equivalent Circuit for Double-Resistive URC Multivibrator

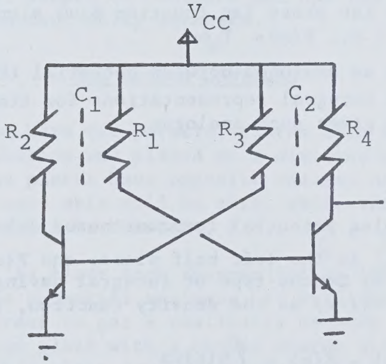

VCC
C1
C2
R2
R1
R3
R4
Vo

Fig. (7) Double-Resistive URC Astable Multivibrator

CONSTANT DELAY NETWORKS - AN INTEGRAL APPROACH

K. S. BAJAJ
Department of Electrical Engineering
Detroit Institute of Technology
Detroit, Michigan 48201

ABSTRACT

A new analogue between Cauchy type of Integrals and potential theory has been used for integral representations of constant delay networks. The types of contours used are linear, circular, elliptic, and parabolic. Network realization of Integral representations are mentioned. Design of constant delay networks, by distributed RC realizable, Cauchy type of integrals has been given.

INTRODUCTION

Constant delay networks have been designed by many authors[1-3]. In this paper, constant delay networks are represented by integrals. Integral representations were recently given for RC, Rl, and LC distributed networks[4]. Integral representations simplify the design of constant delay networks.

1. CONSTANT DELAY NETWORKS

Constant delay networks are characterized by the transfer function[3]

$$H(s) = \epsilon^{-T_D s} \qquad [1]$$

For $s = j\omega$ $H(j\omega) = \epsilon^{-jT_D \omega}$ we see that its magnitude is unity for all real frequencies, and the phase lag function $B(\omega)$ along the j-axis is a linear function of i.e., $B(\omega) = T_D\omega$. [2]

Now, we will use an analogue between potential theory and Cauchy type of integrals[5] to find integral representations for time delay networks. The following theorem gives such analogue.

THEOREM 1

Let W(s) be complex potential for continuous charge distribution -q(-x) on a contour L^- in the left half plane, and F(s) be network function represented by Cauchy type of integral having L^- as the line of discontinuity and -q(-x) as the density function, then:

$$\frac{d}{ds}W(s) = F(s) = \int_{L^+} \frac{q(x)dx}{s+x} \qquad [3]$$

where L^+ is the mirror image of L^- in the right half plane.

PROOF:

Complex potential for a charge q placed at any point s_m is given by:

$$W = -q \log(s-s_m)$$

For a set of point charges, the total potential is simply the sum of individual potentials:

$$W = -\sum q_m \log(s-s_m)$$

Now, for a continuous distribution of charge -q(-x) over a contour L^-, the sum is replaced by an integral:

$$W(s) = -\int -q(-x)\log(s-x)dx \qquad [4]$$

where dx is an element of length on the contour.

From equation 4:

$$\frac{dw(s)}{ds} = \int_{L^-} \frac{-q(-x)dx}{x-s}$$

$$\frac{dw(s)}{ds} = \int_{{}^{+}L} \frac{q(x)dx}{s+x}$$

Now $\frac{dw}{ds} = \frac{\partial V}{\partial x} + \frac{i\partial\psi}{\partial x}$ or

$$\frac{dw}{ds} = \frac{\partial\psi}{\partial y} - \frac{i\,\partial V}{\partial y}$$

where V is the potential function, and ψ is the stream function. Next, we will pick different shapes of contours and use the above given analogue to design constant delay networks.

1.1 LINEAR CONTOURS

Let us suppose we have two parallel plates each having constant density charge distribution and placed at a distance α on each side of the origin. As the plates have opposite charges on them, the field parallel to the imaginary axis will be zero, while the field transverse to the imaginary axis will be constant, i.e., $\frac{dV}{dy} = 0$ and $\frac{d\psi}{dy}$ is constant. Now, it is clear from theorem (1) that if we construct the function from this potential analogue, it will have linear phase characteristic. In order to get a realizable network function, we consider only left side plate with a double charge distribution on it. Taking the trace of this plate as line of discontinuity L^- and the value of charge distribution as density function, we write the function F(s) as:

$$F(s) = \int_L \left[\frac{q(x)dx}{(s+\alpha+ix)} + \frac{q(x)dx}{(s+\alpha-ix)}\right] \qquad [5]$$

$$F(s) = \int_L \frac{2(s+\alpha)\,q(x)dx}{(s+\alpha)^2 + x^2} \qquad [6]$$

In order to have good approximation in the desired band of frequencies, the contour picked should be longer than the frequency band. As an example, the function:

$$F_2(s) = \frac{4}{\pi}\int_0^{1.3} \frac{x(s+\alpha)dx}{(s+\alpha)^2+x^2}$$

For $\alpha = 2$, has a constant magnitude and a linear phase characteristic for $0 \leqslant \omega \leqslant 1$. The response of $F_2(s)$ for $0 \leqslant w \leqslant 1$ is shown in figure (1). F(s) given in equation (5) can be realized by distributed RLC Foster type networks.[5] F(s) can also be realized by RLC ladder network as it can be expanded into a continued fraction of the form:

$$\int_L \frac{2(s+\alpha)\; q(x)dx}{(s+\alpha)^2+x^2} = \frac{1|}{|b_1(s+\alpha)} + \frac{1|}{|b_2(s+\alpha)} + \frac{1|}{|b_3(s+\alpha)} + \frac{1|}{|b_4(s+\alpha)} + \dots$$

Using the algorithm[5] for continued fraction expansion of $F_2(s)$, it has been realized by tapered RLC ladder network, and the network is given in figure (2).

2. CIRCULAR CONTOURS

Now, if we want to design a constant delay network for $-R \leqslant \omega \leqslant R$, we can pick a circular contour of radius R, which embraces the desired frequency range. Left hand side of the contour has positive charge distribution while the right hand side has negative charge distribution. In order to get realizable network functions, we will take only left portion of the contour and double charge density on it. Now, taking the contour as line of discontinuity and charge distribution as density function we have F(s).

$$F(s) = \int_0^{\pi/2} \left[\frac{q(\phi)}{(s+R\mathrm{Cos}\phi + iR\mathrm{Sin}\phi)} + \frac{q(\phi)}{(s+R\mathrm{Cos}\phi - iR\mathrm{Sin}\phi)} \right] d\phi$$

$$F(s) = \int_0^{\pi/2} \frac{2(s+R\mathrm{Cos}\,\phi)\; q(\phi)d\phi}{(s+R\mathrm{Cos}\phi)^2 + R^2\mathrm{Sin}^2\phi}$$

taking charge distribution as $\mathrm{Cos}\phi$

$$F(s) = \int_0^{\pi/2} \frac{2(s+R\;\mathrm{Cos}\phi)\;.\;\mathrm{Cos}\phi\; d\phi}{(s+R\;\mathrm{Cos}\phi)^2 + R^2\mathrm{Sin}^2\phi} \qquad [7]$$

For $R = 1$, function F(s) has constant magnitude and a linear phase characteristic for $0 \leqslant \omega \leqslant 1$. The response of F(s) for $0 \leqslant \omega \leqslant 1$ is shown in figure (3). F(s) can be realized by Foster distributed networks.[5]

ELLIPTIC CONTOURS:

Elliptic distribution[1] of poles have been considered for the design of constant delay networks . We have used Elliptic Contour with positive and negative charge densities on the left and right sides respectively. The equation for ellipse is:

$$\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1 \quad \text{or} \quad \begin{array}{l} x = b\,\mathrm{Cos}\phi \\ y = a\,\mathrm{Sin}\phi \end{array}$$

Once again we will take only the left half portion of the ellipse and double the charge distribution on it. Now, taking the contour as the line of discontinuity and $q(\phi)$, the charge distribution, as density function, we form the network function $F(s)$.

$$F(s) = \int_0^{\pi/2} \left[\frac{q(\phi)}{(s + b\,\mathrm{Cos}\phi + ia\,\mathrm{Sin}\phi)} + \frac{q(\phi)}{(s + b\,\mathrm{Cos}\phi - ia\,\mathrm{Sin}\phi)} \right] d\phi$$

$$F(s) = \int_0^{\pi/2} \frac{2(s + b\,\mathrm{Cos}\phi)\,q(\phi)}{(s + b\,\mathrm{Cos}\phi)^2 + a^2\,\mathrm{Sin}^2\phi}$$

taking $q(\phi) = \mathrm{Cos}\phi$, we have: $$F(s) = \int_0^{\pi/2} \frac{2(s + b\,\mathrm{Cos}\phi)\,\mathrm{Cos}\phi\, d\phi}{(s + b\,\mathrm{Cos}\phi)^2 + a^2\,\mathrm{Sin}^2\phi} \qquad [8]$$

$F(s)$ has a constant magnitude and a linear phase characteristic. These results have been plotted in figure (4) for different $(\frac{b}{a})$ ratios. $F(s)$ can be realized by Foster type distributed networks.[5]

PARABOLIC CONTOURS:

Parabolic[2] distribution of poles have also been considered for the design of constant delay networks. The type of the parabolic contour which we are considering is of the form:

$$y^2 = 4(x + 1) \quad \text{or the parametric equation is} \quad \begin{array}{l} y = 2\mathrm{Cos}\phi \\ x = \mathrm{Sin}^2\phi \end{array}$$

Now, once again taking the contour as the line of discontinuity and the charge distribution as density function, we form the network function as:

$$F(s) = \int_0^{\pi/2} \left[\frac{q(\phi)}{(s - \mathrm{Sin}^2\phi + i2\mathrm{Cos}\phi)} + \frac{q(\phi)}{(s - \mathrm{Sin}^2\phi - i2\mathrm{Cos}\phi)} \right] d\phi$$

$$F(s) = \int_0^{\pi/2} \frac{2q(\phi)\,(s-\sin^2\phi)\,d\phi}{(s-\sin^2\phi)^2 + 4\cos^2\phi} \qquad [9]$$

for $q(\phi) = \cos\phi$, $F(s)$ has a constant magnitude and linear phase characteristic for $0 \leqslant \omega \leqslant 1$. Function $F(s)$ is Foster distributed realizable. Response of $F(s)$ for $0 \leqslant \omega \leqslant 1$ is shown in figure (5).

3. DESIGN OF CONSTANT DELAY NETWORKS BY RC FUNCTIONS

POTENTIAL THEORY APPROACH:

Now, instead of placing the charged conductors parallel to the imaginary axis, they are placed on the real axis at a distance from the origin. Once again we can construct a function having the desired response, from this potential analogue.

The function $F(s)$ is given by: $$F(s) = \int_a^b \frac{q(x)\,dx}{s+x} \qquad [10]$$

Response of the function $F(s) = \int_4^5 \frac{x\,dx}{s+x}$ for $0 \leqslant \omega \leqslant 1$ has been shown in figure(6). $F(s)$ has been realized by tapered distributed RC network as shown in figure (7).

SINGULAR INTEGRAL EQUATIONS APPROACH:

Constant delay networks are characterized by the function:

$H(s) = \epsilon^{-T_D s}$. Now, we want to seek a Cauchy type integral representation for $H(s)$, i.e.,

$$H(s) = \epsilon^{-T_D s} = \int_L \frac{f(x)\,dx}{s+x} \qquad [11]$$

Equation (11) is a singular integral equation and can be solved for $f(x)$. Solution of such integral equations are given in References (6) and (7). Following the approach of Mikhlin[7] we enclose the line interval L by a contour and solve the integral equation: $f(x) = e^{+T_D x}$.

Response of $H(s) = \int_0^{10} \frac{e^{T_D x}\,dx}{s+x}$ for $0 \leqslant \omega \leqslant 1$ is shown in figure (6) and the approximation is very good. The function $H(s)$ can be realized by distributed RC networks.

REFERENCES

1. Scanlan, J. O., "Transfer Functions with Elliptic Distribution of Poles at Equal Spacings." I.E.E.E. Transactions on Circuit Theory, Vol. CT-12, pp. 206-266, June 1965.

2. Rakovich, B. D.and Rabrenovich, D. M., "Transfer Functions With Poles Located on Parabolic and Elliptic Contours," Electronics Letters, Vol. 3, pp. 201-202, May 1967.

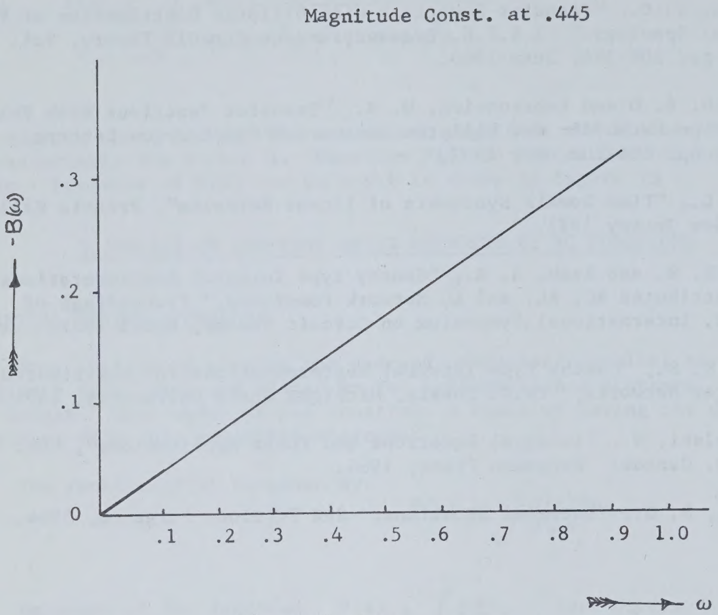
3. Su, K. L., "Time Domain Synthesis of Linear Networks", Prentic Hall, Inc., New Jersey 1971.

4. Bajaj, K. S. and Resh, J. A., "Cauchy type Integral Representations for Distributed RC, RL, and LC Network Functions," Proceedings of I.E.E.E. International Symposium on Circuit Theory, April 1972.

5. Bajaj, K. S., "Cauchy Type Integral Representations for Distributed Parameter Networks," Ph.D. Thesis, Michigan State University, 1971.

6. Pogorzelski, W., "Integral Equations and Their Applications", Vol. 1, Toronto, Canada: Pergamon Press, 1966.

7. Mikhlin, S. G., "Integral Equations," 2nd Edition, Pergamon, 1964.



Linear Contour

fig (1)

Circular Contour

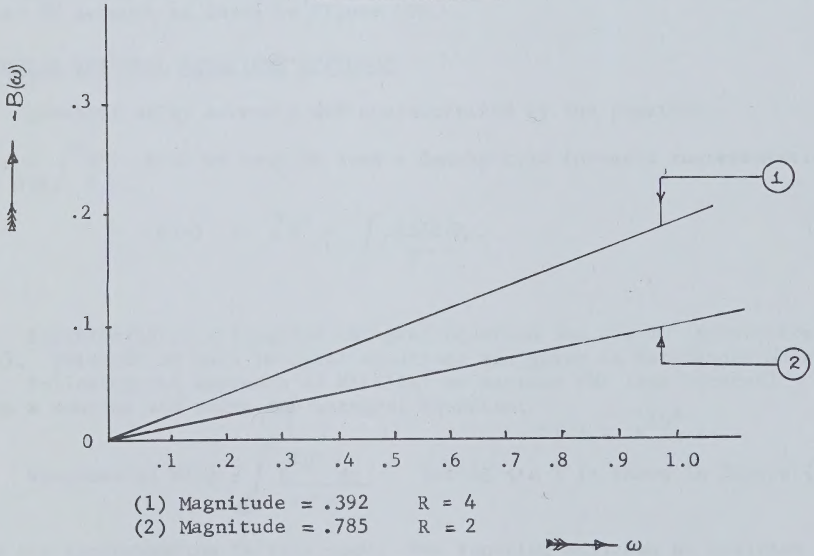
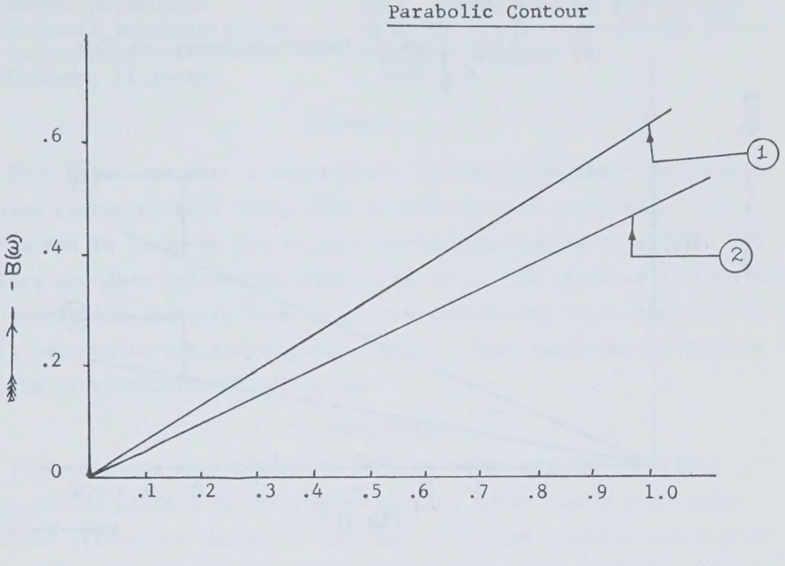
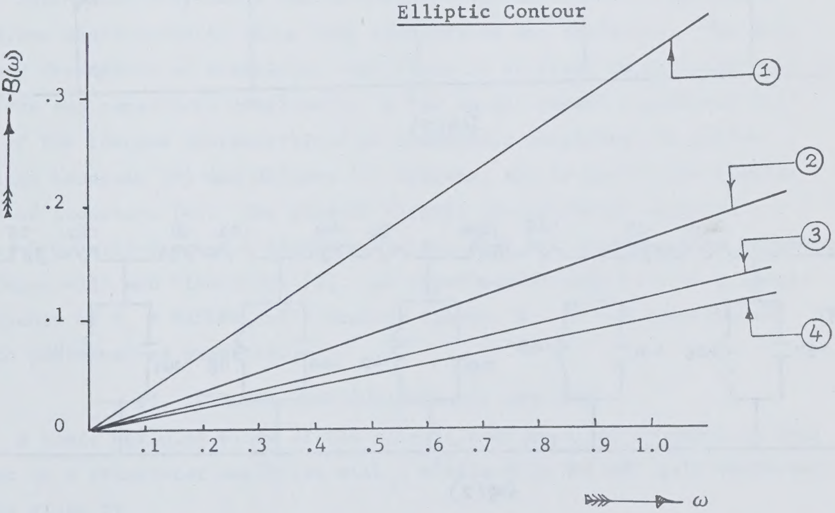


fig (3)



(1) q (ϕ) = Cosϕ Magnitude = 1.66
(2) q (ϕ) = 2.0 Magnitude = 2.10
fig (5)



(1) b/a = 1
(2) b/a = 2
(3) b/a = 4.0
(4) b/a = 5.0
fig (4)

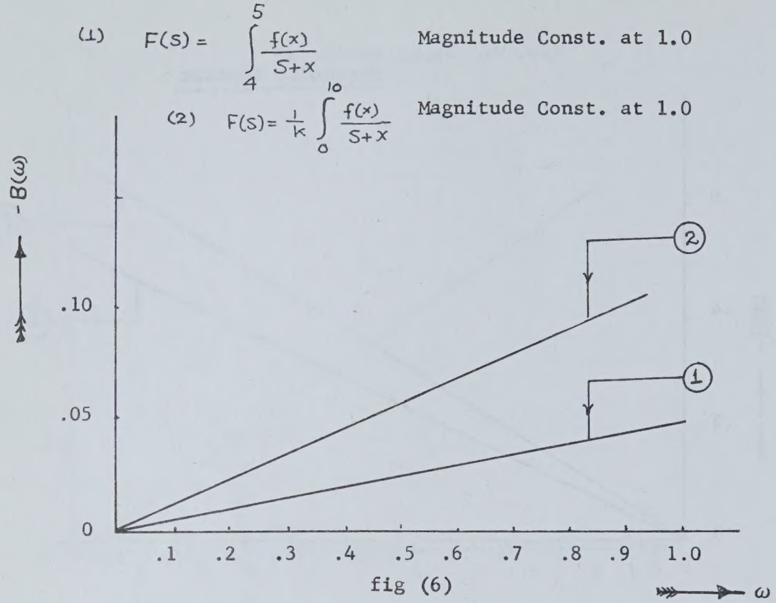
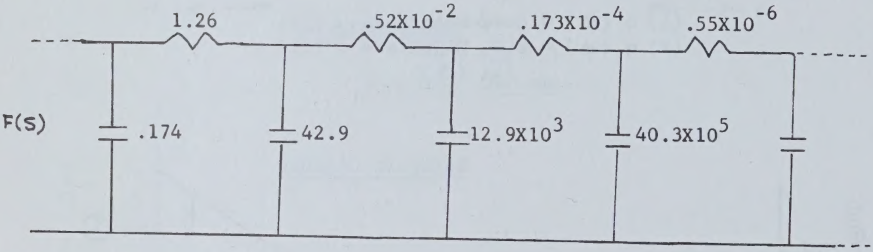
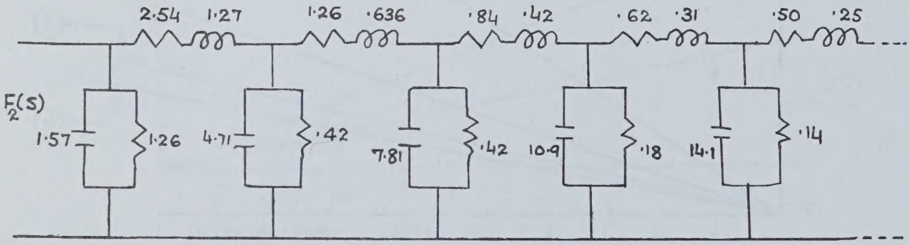


fig (6)


1.26
.52X10^{-2}
.173X10^{-4}
.55X10^{-6}
F(S)
.174
42.9
12.9X10^{3}
40.3X10^{5}

fig(7)


2·54
1·27
1·26
·636
·84
·42
·62
·31
·50
·25
$F_2(S)$
1·57
1·26
4·71
·42
7·81
·42
10·9
·18
14·1
·14

fig(2)

ACTIVE LC-LESS BANDPASS FILTER USING TRANSISTOR LOWPASS CHARACTERISTIC

GEORGE A. PAPHITIS
Engrg. Department
National Broadcasting Co.
Merchandise Mart Plaza
Chicago, Illinois

TADAO MURATA
Dept. of Information Engineering
Univ. of Illinois at Chicago Circle
Chicago, Illinois

ABSTRACT

This paper presents a new circuit configuration which realizes a bandpass characteristic using only transistors and resistors. The realization is based on the lowpass characteristics of transistor amplifiers and does not use LC elements at all. The proposed circuit has many attractive features such as low sensitivities, high stability, and easily adjustable complex-conjugate poles. Some experimental results with the proposed circuit are given.

I. INTRODUCTION

It is obvious that inductors must be eliminated or simulated to design active filters in integrated circuit forms, and a tremendous amount of literature exists in the area of active inductorless filter design [1]. Presently, capacitors in integrated circuits are available only with restricted values and wide tolerances, and require more space than transistors and resistors. Thus, it is also desirable to eliminate or simulate capacitors in active integrated filter realizations.

This paper proposes a new circuit configuration which realizes a bandpass characteristic using only transistors and resistors. The frequency dependence of transistor amplifiers is utilized to eliminate inductors and capacitors completely. A few recent papers considered the use of the lowpass characteristic of transistor amplifiers in active LC-less bandpass [2] and allpass [3] filters, and in the C-less simulation of inductors [4]. The present circuit configuration achieves a lower Q-sensitivity and a higher stability than the configuration proposed by Capparelli and Liberatore [2]. An experimental circuit with a center frequency of f_o = 935kHz and a quality factor, Q = 85 was constructed, which performed as predicted.

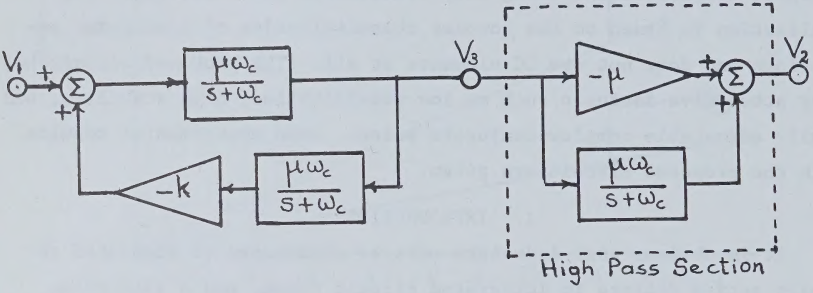
II. CIRCUIT CONFIGURATION AND ANALYSIS

A basic building block of the circuit configuration proposed in this paper is a transistor amplifier with a single-pole rolloff gain characteristic given by

$$A(s) = \frac{\mu\omega_c}{s + \omega_c} \qquad (1)$$

where ω_c is the cutoff frequency and μ is the amplification factor for frequencies less than ω_c. The feedback configuration using two basic blocks and one frequency-independent amplifier (-k) as shown by the first section in Fig. 1. yields the following transfer function from V_1 to V_3

$$T_1 = \frac{V_3}{V_1} = \frac{\mu\omega_c(s+\omega_c)}{s^2+2\omega_c s+(1+k\mu^2)\omega_c^2} \quad . \tag{2}$$



V_1
+
Σ
+
$\frac{\mu\omega_c}{s+\omega_c}$
V_3
-k
$\frac{\mu\omega_c}{s+\omega_c}$
$-\mu$
+
Σ
+
$\frac{\mu\omega_c}{s+\omega_c}$
V_2
High Pass Section

Fig. 1. Block Diagram of a Second-Order Bandpass Filter

T_1 has a finite zero at $s = -\omega_c$ and a complex-conjugate pair of poles at $s = -\omega_c \pm j\omega_c\mu\sqrt{k}$. If $\omega_c\mu\sqrt{k} >> \omega_c$, i.e. for a high quality factor $\mu\sqrt{k}$ is much greater than unity, then the pole-zero pattern of T_1 becomes approximately equal to that of the second-order bandpass function. This was checked by experiments, and it was found that T_1 was indeed a good approximate bandpass function for higher quality factors, i.e. $Q > 30$.

An exact bandpass function can be obtained by connecting in cascade a first-order highpass section as is shown in Fig. 1. This highpass section has the transfer function from V_3 to V_2 given by

$$T_2 = \frac{V_2}{V_3} = \frac{-\mu s}{s+\omega_c} \quad . \tag{3}$$

Thus, the overall transfer function from V_1 to V_2 is given by the following second-order bandpass function

$$T = \frac{V_2}{V_1} = T_1 \; T_2 = \frac{-\mu^2\omega_c s}{s^2+2\omega_c s+(1+k\mu^2)\omega_c^2} \quad , \tag{4}$$

which has a center frequency given by

$$\omega_o = \omega_c \sqrt{1 + k\mu^2} \quad (5)$$

and a quality factor at ω_o given by

$$Q = \sqrt{(1 + k\mu^2)/4} \quad (6)$$

The differential sensitivities of Q with respect to k and μ are easily derived from (6), respectively, as follows:

$$S_k^Q = \frac{k\mu^2}{2(1 + k\mu^2)} < \frac{1}{2} \text{ and } S_\mu^Q = \frac{k\mu^2}{1 + k\mu^2} < 1 \quad .$$

Note that the above Q- sensitivities are low (comparable to a passive network). However, the configuration given in [2] yields $S_k^Q = 2Q - 1$, which means that $S_k^Q = 39$ even for a low $Q = 20$ case. In addition, the poles of the present configuration given by $s = -\omega_c \pm j\omega_c\mu\sqrt{k}$ are always in the left-half plane (stable) and can easily be adjusted by ω_c, k and μ. Also, Q can be controlled by k and μ. The center frequency ω_o is given by $\omega_o \cong k\mu\omega_c$ for $k\mu^2 >> 1$ and is practically independent of ω_c and μ for a circuit with a constant gain-bandwidth product ($\mu\omega_c$ = constant).

III. EXPERIMENTAL RESULTS

To test the configuration presented here, an experimental circuit was constructed. For the basic building block characterized by (1), the transistor amplifier circuit shown in Fig. 2(a) was chosen. The common-base transistor in Fig. 2(a) provides frequency-dependent amplifcation, where the cutoff frequency $f_c = {}^{\omega}c/2\pi$ can be controlled by the two bias resistors R_1 and R_2. With this circuit, it was possible to set f_c between 5.5kHz and 150kHz. However, f_c can be adjusted beyond this range by using different transistors. Fig. 2(b) shows the experimentally observed frequency characteristic of the circuit for f_c = 20kHz along with the ideal single-pole rolloff characteristic.

Fig. 3 is a second-order bandpass filter which was built according to the block diagram in Fig. 1, with center frequency of f_o = 935kHz and quality factor Q = 85, where k = 480, μ = 7.7 and f_c = 5.5kHz. For this circuit, a clear resonance curve as shown in Fig. 4 was observed with the upper half-power frequency f_2 = 941kHz and the lower half-power frequency f_1 = 930kHz.

Experiments were also performed on the approximate bandpass function $T_1 = V_3/V_1$. A clear lowpass characteristic such as that which is shown in Fig. 5 could not be observed until the quality factor was lowered to Q=21. Fig. 5 shows the observed frequency characteristic of T_1 for Q = 21, where the lowpass curve is about 40 db below the peak resonance point.

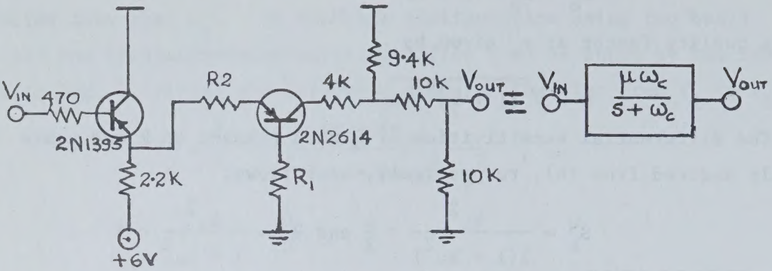

-6V
-6V
9·4K
VIN
470
R2
4K
10K
VOUT
≡
VIN
$\frac{\mu\omega_c}{s+\omega_c}$
VOUT
2N1395
2N2614
2·2K
R1
10K
+6V

Fig. 2(a) Transistor Amplifier Used for a Single-Pole Rolloff Characteristic.

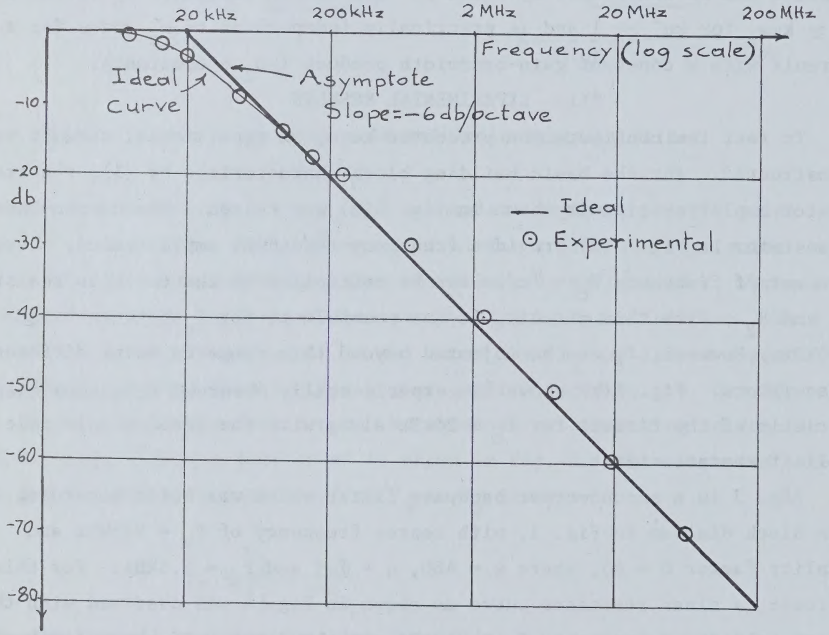


Fig. 2(b) Frequency Characteristics of the Amplifier

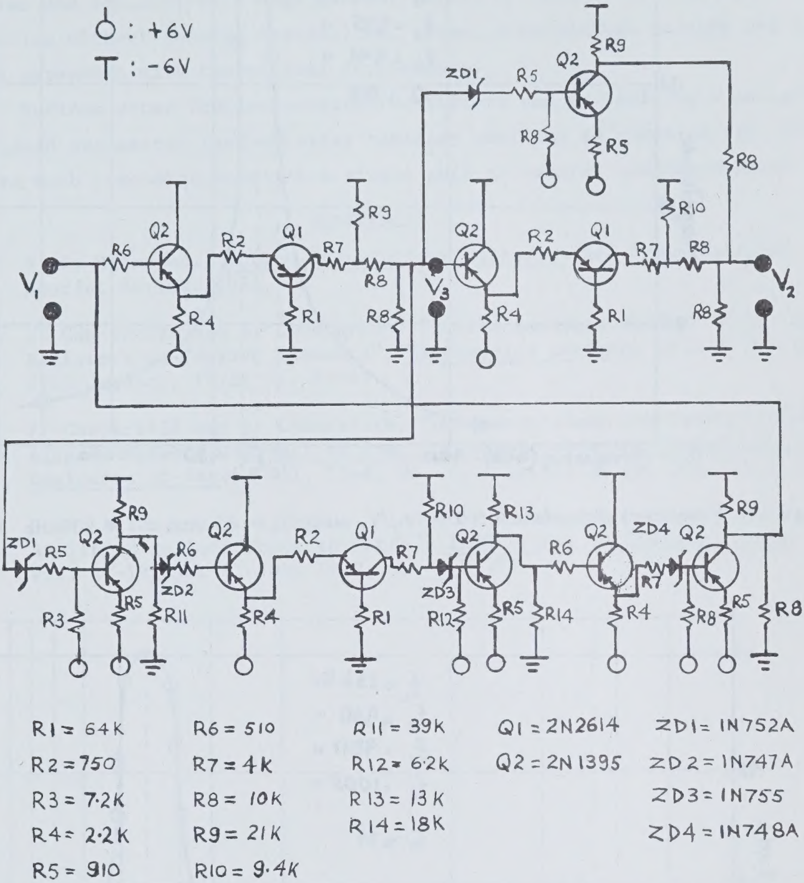

: +6V
T : -6V
V1
V2
V3
Q1
Q2
ZD1
ZD2
ZD3
ZD4
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R1 = 64K
R2 = 750
R3 = 7.2K
R4 = 2.2K
R5 = 910
R6 = 510
R7 = 4K
R8 = 10K
R9 = 21K
R10 = 9.4K
R11 = 39K
R12 = 6.2K
R13 = 13K
R14 = 18K
Q1 = 2N2614
Q2 = 2N1395
ZD1 = 1N752A
ZD2 = 1N747A
ZD3 = 1N755
ZD4 = 1N748A

Fig. 3. Experimental Circuit of Fig. 1.

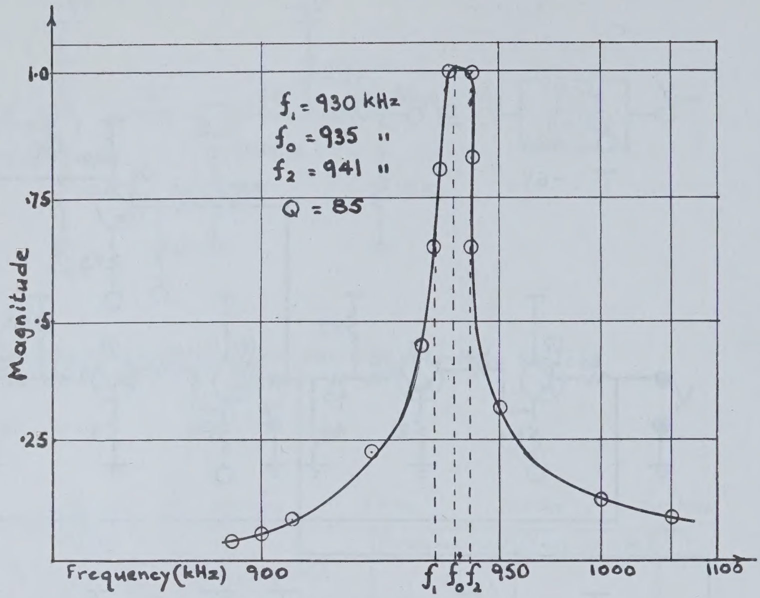


Fig. 4. Frequency Response of $T = V_2/V_1$ with $Q = 85$ and $f_o = 935\text{kHz}$

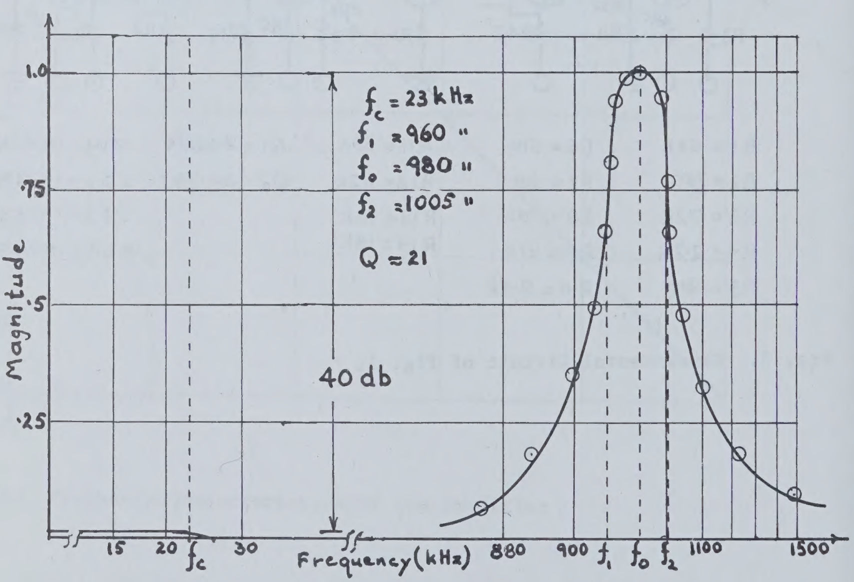


Fig. 5. Frequency Response of Approximate Bandpass Function, $T_1 = V_3/V_1$ with $Q = 21$.

IV. CONCLUSION

An active LC-less bandpass filter suitable for integrated circuit fabrication has been presented. The proposed circuit has low sensitivities and can achieve a high quality factor at relatively high frequencies without causing instability. Also, experimental results are in good agreement with theoretical performance.

Various other frequency-selective filters can be built by a cascading isolated resonators (second-order bandpass sections as shown in Fig. 1), where each resonator supplies a single pair of complex conjugate poles.

REFERENCES

[1] S. K. Mitra(ed.) Active Inductorless Filters, IEEE Press Selected Reprint Series, 1971.

[2] F. Capparelli and A. Liberatore, "Active Bandpass Network With Only Resistors as Passive Elements", Electronics Letters, Vol. 8, No. 2, 27th January, 1972, pp. 43-44.

[3] F. Capparelli and A. Liberatore, "An Inductorless and Capacitorless Allpass Network", Trans. of Inst. of Electronics and Communication Engineers of Japan, Vol. 55-A, No. 7, July, 1972, pp. 304-365.

[4] P. E. Allen and J. A. Means, "Inductor Simulation Derived from an Amplifier Rolloff Characteristic", IEEE Trans. on Circuit Theory, Vol. CT-19, No. 4, July 1972, pp. 397-398.

SYNTHESIS OF TWO-ELEMENT-KIND CASCADE NETWORKS

TING N. LEE
Department of Electrical Engineering and Computer Science
The George Washington University
Washington, D. C. 20006

DAVID P. BROWN
Department of Electrical Engineering
The University of Wisconsin
Madison, Wisconsin 53706

ABSTRACT

Analysis and synthesis of certain two-element-kind cascaded two-ports is investigated in this paper. Starting with a matrix which is to be the cascade matrix of some two-port, procedures to decompose the matrix into a product of matrices each of which is a cascade matrix of a simple two-port are given. The decomposition procedure presented here is straightforward and the necessary and sufficient conditions on the entries in the matrix for the process to work are established.

The results of Weinberg[1] are related to the material in this paper. In addition, algorithms have been used by Balabanian[2] and Hunt[3] for synthesis of two-ports from cascade matrices.

ZEROS OF CASCADE PARAMETERS

Consider the cascaded network in Figure 1(a) where each two-port is characterized by cascade parameters:

$$\begin{bmatrix} V_{i+1} \\ I_{i+1} \end{bmatrix} = \begin{bmatrix} A_i(s) & B_i(s) \\ C_i(s) & D_i(s) \end{bmatrix} \begin{bmatrix} V_i \\ I_i \end{bmatrix} = K_i(s) \begin{bmatrix} V_i \\ I_i \end{bmatrix}$$

Thus, the overall cascade parameter matrix K(s) is simply the product of the individual cascade parameter matrices,

$$K(s) = K_n(s)\, K_{n-1}(s) \quad \cdots \quad K_1(s).$$

In the following two properties the zeros of the cascade parameters for certain two-element-kind networks are characterized.

Property 1 - Consider any cascaded network where each two-port is as shown in Fig. 1(b) or each two-port is as shown in Fig. 1(c). If A(s), B(s) C(s), D(s) are the overall cascade parameters, then

(a) A(s), B(s), C(s) and D(s) are polynomials of s,

(b) the zeros of A(s) and C(s), B(s) and D(s), A(s) and B(s), C(s) and D(s) interlace on the negative real axis,

(c) the degree of A(s) and B(s), B(s) and D(s), C(s) and D(s), A(s) and C(s) cannot differ by more than one.

Proof: The cascade parameter matrix for one section is, by simple analysis,

$$K_i = \begin{bmatrix} 1+(L_i s+R_i)G_i & L_i s+R_i \\ G_i & 1 \end{bmatrix} \quad \text{for Fig. 1(b)}$$

or

$$K_i = \begin{bmatrix} 1+R_i(G_i+sC_i) & R_i \\ C_i s+G_i & 1 \end{bmatrix} \quad \text{for Fig. 1(c).}$$

Then since the overall cascade matrix is simply the product of the K_i's, (a) of the conclusion follows. The driving point impedance of a cascaded network is

$$Z(s) = \frac{A(s)Z_O + B(s)}{C(s)Z_O + D(s)}$$

where $Z_O=V_O/I_O$ is any load impedance. In particular, let $Z_O=0$. Since it is known that the driving point impedance Z(s) of any RC or RL network has alternating zeros and poles on the negative real axis, this implies the zeros of A(s) and C(s) interlace on the negative real axis. By letting $Z_O=\infty$, it follows that zeros of B(s) and D(s) interlace on the negative real axis.

By reversing the two current polarities of each two-port of Fig. 1(a) the output impedance of the cascaded network is

$$Z_{oi}(s) = \frac{D(s)Z_O + B(s)}{C(s)Z_O + A(s)}$$

where Z_O is the termination at the input. Since $Z_{oi}(s)$ is the driving point impedance of an RC or RL network (b) follows by applying the above argument.

Conclusion (c) follows since the numerator and denominator of the driving point impedance of any RC or RL network cannot differ by more than one degree.

Consider a cascaded network in which each two-port is shown in Fig.1(d). It can be established that the cascade parameters of this network satisfy: A(s) and D(s) are even polynomials and B(s) and C(s) are odd polynomials, (b) of property 1 with negative real axis replaced by imaginary axis and (c) of property 1.

In the next property additional interlacing zero characteristics are established using the assumption, which is satisfied by reciprocal two-ports, that the determinant of the cascade parameter matrix is unity.

Property 2 - Suppose det.$\begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix} = 1$, A(s) and B(s) are polynomials of degree n, and C(s) and D(s) are polynomials of degree n-1, where $n>1$. If zeros of A(s) and C(s) are real and interlacing, then the pairs of polynomials A(s) and B(s), B(s) and D(s), and C(s) and D(s) have real and interlacing zeros.

Proof: (a) Let z_{a_i} and $z_{a_{i+1}}$ be the ith and i+1th zeros of A(s) respectively. By hypothesis, $A(s)D(s)-B(s)C(s)=1$.
Hence, $B(z_{a_i})C(z_{a_i})=B(z_{a_{i+1}})C(z_{a_{i+1}})=1$. Since zeros of A(s) and C(s) interlace on the real axis, $C(z_{a_i})$ and $C(z_{a_{i+1}})$ are nonzero and of opposite sign. It follows that $B(z_{a_i})$ and $B(z_{a_{i+1}})$ are nonzero and of opposite sign. This implies that the interval $(z_{a_i},z_{a_{i+1}})$ contains an odd number of zeros of B(s). However, there are n-1 of these intervals, therefore each contains one and only one zero of B(s). The remaining one zero of B(s) can be located either to the right of the z_{a_1} or to the left of the z_{a_n}. This gives (a) of conclusion.

To establish (b) and (c) arguments similar to the one given above are used.

Additional interlacing zero properties are established as in Property 2 using different polynomial degrees and/or different assumed interlacing zeros. The following table gives some additional interlacing zero patterns. In each case it is assumed that the determinant of the cascade parameter matrix is unity.

A(s)	B(s)	C(s)	D(s)	Interlacing Zero Property
n	n	n	n	C,D (or A,B) → A,C; B,D; A,B (or C,D)
n	n	n-1	n-1	A,B → A,C; B,D; C,D
n	n-1	n	n-1	A,C → A,B; B,D; C,D
n-1	n	n	n-1	C,D → A,B; B,D; A,C
2n	2n-1	2n-1	2(n-1)	A,C → A,B; B,D; C,D
2(n-1)	2n-3	2n-1	2(n-1)	C,D → A,C; A,B; B,D

A row in the table gives the degrees of the polynomials and the corresponding interlacing property, for example the interlacing of A and B implies the interlacing of A and C, B and D, and C and D (see row 2).

DECOMPOSITION OF CASCADE MATRICES

In the following new theorems, a procedure is established to decompose matrices containing polynomial entries satisfying zero pattern restrictions. The form of the matrices in the decomposition is appropriate for cascade network synthesis.

The first theorem and the corollaries which follow it pertain to R-L network decomposition procedures.

Theorem 1 Let the matrix $K(s)=\begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix}$. If and only if

(a) Det. $K(s)=1$,

(b) $A(s)$, $B(s)$, $C(s)$ and $D(s)$ are polynomials with positive coefficients except for the constant terms which may vanish, $A(s)$ and $B(s)$ are of degree n, $C(s)$ and $D(s)$ are of degree n-1, $n>1$,

(c) the zeros of $A(s)$ and $C(s)$ interlace on the negative real axis with no zero of $A(s)$ at the origin,

then there exists a decomposition such that

$$K(s) = K_n K_{n-1} \cdots K_1 K_0 \quad , \text{ where}$$

$$K_i = \begin{bmatrix} 1+s\ell_i g_i & s\ell_i \\ g_i & 1 \end{bmatrix} , \quad g_i, \ell_i > 0,$$

and

$$K_0 = \begin{bmatrix} a_0 & b_0 \\ 0 & 1/a_0 \end{bmatrix} , \quad a_0 = A(0) \neq 0, \quad B(0) = b_0 .$$

Proof: Sufficiency : Suppose K(s) of hypothesis is

$$K(s) = \begin{bmatrix} a_n s^n + a_{n-1}s^{n-1} + \dots + a_0 & b_n s^n + b_{n-1}s^{n-1} + \dots + b_0 \\ c_{n-1}s^{n-1} + c_{n-2}s^{n-2} + \dots + c_0 & d_{n-1}s^{n-1} + d_{n-2}s^{n-2} + \dots + d_0 \end{bmatrix}$$

where a_i, b_i, c_i, $d_i > 0$, for all $i \neq 0$.

Define

$$K_n(s) = \begin{bmatrix} 1+s\ell_n g_n & s\ell_n \\ g_n & 1 \end{bmatrix} , \text{ where } \ell_n = \frac{a_n}{c_{n-1}} \text{ and } g_n = \frac{c_{n-1}^2}{a_{n-1}c_{n-1} - a_n c_{n-2}} .$$

Since a_n and c_{n-1} are positive by hypothesis, $\ell_n > 0$. But the sum of the zeros of $A(s)$ is larger than the sum of the zeros of $C(s)$, hence

$$\frac{a_{n-1}}{a_n} - \frac{c_{n-2}}{c_{n-1}} > 0 . \quad \text{This implies } g_n > 0.$$

Observe that

$$K_n(s)^{-1} = \begin{bmatrix} 1 & -s\ell_n \\ -g_n & 1+s\ell_n g_n \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -g_n & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -s\ell_n \\ 0 & 1 \end{bmatrix}.$$

The product

$$K_n(s)^{-1}K(s) = \begin{bmatrix} A'\ s) & B'(s) \\ C'(s) & D'(s) \end{bmatrix}$$

where

$$A'(s) = s^n(a_n - c_{n-1}\ell_n) + s^{n-1}(a_{n-1} - c_{n-2}\ell_n) + \dots + a_0$$
$$B'(s) = s^n(b_n - d_{n-1}\ell_n) + s^{n-1}(b_{n-1} - d_{n-2}\ell_n) + \dots + b_0$$
$$C'(s) = s^n(g_n c_{n-1}\ell_n - a_n g_n) + s^{n-1}(-g_n a_{n-1} + c_{n-2}\ell_n g_n + c_{n-1}) + \dots$$
$$D'(s) = s^n(-b_n g_n + g_n d_{n-1}\ell_n) + s^{n-1}(-g_n b_{n-1} + d_{n-2}\ell_n g_n + d_{n-1}) + \dots$$

The coefficient of s^n in $A'(s)$ vanishes by the definition of ℓ_n. A detailed calculation of the coefficient relations required by (a) of hypothesis implies that the coefficient of s^n in $B'(s)$ is zero. The coefficient of s^n and s^{n-1} of $C'(s)$ vanishes by definition of ℓ_n and g_n. This is also true for the coefficient of s^n and s^{n-1} in $D'(s)$. It follows from (c) of hypothesis that

$$\frac{a_i}{a_n} - \frac{c_{i-1}}{c_{n-1}} > 0, \quad i \neq 0.$$

Therefore it follows that the coefficient of s^i in $A'(s)$ is positive. The same result can be established for the coefficients of $B'(s)$.

Next consider

$$K_n^{-1}K(s) = \begin{bmatrix} 1 & 0 \\ -g_n & 1 \end{bmatrix} \begin{bmatrix} 1 & -s\ell_n \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -g_n & 1 \end{bmatrix} \begin{bmatrix} A'(s) & B'(s) \\ C\ (s) & D\ (s) \end{bmatrix}.$$

The entries of the matrix $\begin{bmatrix} A'(s) & B'(s) \\ C\ (s) & D\ (s) \end{bmatrix}$ are polynomials of the same degree, n-1 and its determinant is 1. By hypothesis and Property 2, zeros of $C(s)$ and $D(s)$ interlace on the negative real axis. Since the degree of $A'(s)$ and $B'(s)$ is n-1, hence the degree of $C'(s)$ and $D'(s)$ is $\geq$ n-2. This implies that the degree of $C'(s)$ and $D'(s)$ is n-2, since degree $C'(s)$ and $D'(s)$ is also $\leq$n-2. Hence by the second row of Table, it follows that zeros of $A'(s)$ and $C'(s)$ interlace on the negative real axis. Therefore, the matrix $K_n^{-1}(s)K(s)$ possesses properties (a), (b) and (c) with every entry one degree less than that of the $K(s)$.

The foregoing decomposition process is continued until the following matrix results,

$$\begin{bmatrix} a_1 s + a_0 & b_1 s + b_0 \\ c_0 & d_0 \end{bmatrix},$$

where $a_1 d_0 - b_1 c_0 = 0$ and $a_0 d_0 - c_0 b_0 = 1$.

Define

$$K_1(s) = \begin{bmatrix} 1+s\ell_1 g_1 & s\ell_1 \\ g_1 & 1 \end{bmatrix}$$

where

$$\ell_1 = \frac{a_1}{c_0} \quad \text{and} \quad g_1 = \frac{c_0}{a_0}.$$

Therefore, the product

$$\begin{bmatrix} 1 & -s\ell_1 \\ -g_1 & 1+s\ell_1 g_1 \end{bmatrix} \begin{bmatrix} a_1 s + a_0 & b_1 s + b_0 \\ c_0 & d_0 \end{bmatrix} = \begin{bmatrix} a_0 & b_0 \\ 0 & 1/a_0 \end{bmatrix} = K_0$$

where $A(0)=a_0\neq 0$ and $B(0)=b_0$. This completes the proof of the sufficiency part of the theorem.

Necessity: This is established by induction but the details are omitted.

Q.E.D.

Corollary 1 If and only if

(a) Det. $K(s)=1$,

(b) $A(s)$, $B(s)$, $C(s)$, $D(s)$ are polynomials with positive coefficients and of the same degree n,

(c) $A(s)$ and $C(s)$ have zeros interlaced on the negative real axis with the first zero of $A(s)$ nearest the origin, then there exists a decomposition such that

$$K(s) = K_n' K_n K_{n-1} \cdot K_1 K_0$$

where $K_i = \begin{bmatrix} 1+s\ell_i g_i & s\ell_i \\ g_i & 1 \end{bmatrix}$, $\ell_i, g_i > 0$ for $i\neq 0$ and $K_0 = \begin{bmatrix} a_0 & b_1 \\ 0 & 1/a_0 \end{bmatrix}$

and

$$K_n' = \begin{bmatrix} 1 & 0 \\ g_{n+1} & 1 \end{bmatrix} .$$

Corollary 2 If and only if

(a) Det. $K(s)=1$,

(b) $A(s)$, $B(s)$, $C(s)$, $D(s)$ are polynomials with positive coefficients, $A(s)$ and $B(s)$ are of degree n, $C(s)$ and $D(s)$ are of degree n-1, for $n>1$,

(c) the zeros of $A(s)$ and $B(s)$ interlace on the negative real axis, with a zero of $B(s)$ nearest or at the origin,

then there exists a decomposition such that

$$K(s) = \bar{K}_0 \bar{K}_1 \ldots \bar{K}_{n-1} \bar{K}_n$$

where

$$K_i(s) = \begin{bmatrix} 1+s\bar{\ell}_i \bar{g}_i & s\bar{\ell}_i \\ \bar{g}_i & 1 \end{bmatrix} \text{ for all } i>0, \text{ and } K_0 = \begin{bmatrix} 1/\bar{d}_0 & \bar{b}_0 \\ 0 & \bar{d}_0 \end{bmatrix}$$

where $D(0)=\bar{d}_0\neq 0$, and $B(0)=\bar{b}_0$.

Proof: The proof of this corollary is similar to that of the theorem, only the decomposition process starts from the right hand side of the matrix $K(s)$.

The next theorem and the corollary which follows it pertain to RC network decomposition procedures.

Theorem 2 Let the matrix $K(s) = \begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix}$. If and only if,

(a) Det. $K(s)=1$,

(b) $A(s)$, $B(s)$, $C(s)$, $D(s)$ are polynomials with positive coefficients, $A(s)$ and $C(s)$ are of degree n, $B(s)$ and $D(s)$ are of degree n-1 for $n>1$,

(c) $A(s)$ and $C(s)$ have interlacing zeros on the negative real axis with a zero of $C(s)$ nearest to or at the origin, i.e.,

$$0 \geq z_{c_1} > z_{a_1} > \ldots z_{c_n} > z_{a_n} ,$$

then there exists a decomposition such that

$$K(s) = K_n K_{n-1} \ldots K_1 K_0 ,$$

where

$$K_i = \begin{bmatrix} 1+sc_i r_i & r_i \\ sc_i & 1 \end{bmatrix}, \quad r_i\ , c_i > 0 \ \text{ for } i > 1, \text{ and}$$

$$K_0 = \begin{bmatrix} 1/d_0 & 0 \\ c_0 & d_0 \end{bmatrix}, \quad c_0 = C(0),\ d_0 = D(0) \neq 0.$$

Proof: The proof of this theorem is similar to that of Theorem 1 and is therefore omitted.

Corollary 3 If and only if

(a) Det. $K(s)=1$,

(b) $A(s)$, $B(s)$, $C(s)$, $D(s)$ are polynomials with positive coefficients, $A(s)$ and $D(s)$ are of degree n, $B(s)$ and $C(s)$ are of degree n-1 and n+1 respectively,

(c) $A(s)$ and $C(s)$ have interlacing zeros on the negative axis with a first zero of $C(s)$ nearest to the origin,

then there exists a decomposition such that

$$K(s) = K'_n K_n K_{n-1} \cdots K_1 K_0\ ,$$

where

$$K_i = \begin{bmatrix} 1+s\bar{c}_i\bar{r}_i & \bar{r}_i \\ s\bar{c}_i & 1 \end{bmatrix}, \quad \bar{r}_i,\ \bar{c}_i > 0 \text{ for } i>0,$$

$$K'_n = \begin{bmatrix} 1 & 0 \\ s\bar{c}'_n & 1 \end{bmatrix}, \quad \bar{c}'_n > 0, \text{ and}$$

$$K_0 = \begin{bmatrix} 1/d_0 & 0 \\ c_0 & d_0 \end{bmatrix}, \quad D(0)=d_0 \neq 0, \quad C(0)=c_0.$$

EXAMPLES AND EXTENSION

Given the matrix,

$$S_4(s) = \begin{bmatrix} 2304s^4+5008s^3+2260s^2+265s+4 & 2340s^4+5082s^3+2288.75s^2+262.5s+4 \\ 2304s^3+7204s^2+708s+40 & 2340s^3+2742s^2+716.75s+40.25 \end{bmatrix}$$

It can be shown that conditions (a), (b) and (c) of Theorem 1 are satisfied. Therefore

$$\ell_4 = a_4/c_3 = 2304/2304 = 1, \qquad g_4 = 2304/2304 = 1.$$

and thus after the first step of decomposition,

$$S_4(s) = \begin{bmatrix} 1+s & s \\ 1 & 1 \end{bmatrix} \cdot S_3(s), \text{ where}$$

$$S_3(s) = \begin{bmatrix} 2304s^3+1552s^2+22s+4 & 2340s^3+1572s^2+222.25s+4 \\ 1152s^2+448s+36 & 1170s^2+494.5s+36.25 \end{bmatrix}.$$

Repeating the above procedure on $S_3(s)$ obtains $\ell_2=2$, $g_2=2$ and continuing the complete decomposition is obtained.

$$S_4(s) = \begin{bmatrix} 1+s & s \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1+4s & 2s \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1+9s & 3s \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1+16s & 4s \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 \\ 0 & 1/4 \end{bmatrix}.$$

Fig. 2 shows the corresponding RL network.

The following matrix

$$S_3(s) = \begin{bmatrix} 12s^3+30s^2+14s+1 & 6s^2+12s+4 \\ 12s^3+26s^2+6s & 6s^2+12s+1 \end{bmatrix}$$ satisfies the hypothesis of Theorem 2 and therefore the decomposition gives $r_3=a_3/c_3=1$ and $\tau_3=c_3^2/a_2c_3-a_3c_2= 3$ etc. Therefore,

$$S_3(s) = \begin{bmatrix} 1+3s & 1 \\ 3s & 1 \end{bmatrix} \begin{bmatrix} 1+2s & 2 \\ s & 1 \end{bmatrix} \begin{bmatrix} 1+2s & 1 \\ 2s & 1 \end{bmatrix},$$

and Fig. 3 shows the corresponding RC network.

In the following discussion, a transformation: $s=Z(p)$, such that $Z(p)$ is p.r. function of p, is introduced.

Suppose a matrix

$$K(z) = \begin{bmatrix} A(Z) & B(Z) \\ C(Z) & D(Z) \end{bmatrix}$$ is given with the following properties:

(a) $A(Z)D(Z) -B(Z)C(Z)=1$,

(b) $Z(p)$ is a positive real function of p,

(c) $A(Z)$, $B(Z)$, $C(Z)$, $D(Z)$ are polynomials of Z with positive coefficients except the constant terms which may vanish, $A(Z)$ and $B(Z)$ are of degree n, $C(Z)$ and $D(Z)$ are of degree n-1, for $n>1$,

(d) the zeros of $A(Z)$ and $C(Z)$ interlace on the negative real axis of the Z-plane.

Then by Theorem 1, a decomposition exists in the following form,

$$K(Z) = K_nK_{n-1}\cdots K_1K_0,$$

where

$$K_i = \begin{bmatrix} 1+Z\ell_i g_i & Z\ell_i \\ g_i & 1 \end{bmatrix}$$ for all $i>0$, and

$$K_0 = \begin{bmatrix} a_0 & b_0 \\ 0 & 1/a_0 \end{bmatrix}, \quad a_0=A(Z)\big|_{Z=0}>0, \quad b_0=B(Z)_{Z=0} \quad .$$

REFERENCES

1. Weinberg, L. "Network Analysis and Synthesis", McGraw-Hill Book Co., N.Y., pp 334-342, 1962.
2. Balabanian, N. "Network Synthesis", Prentice-Hall, N.J., pp. 343-352, 1958.
3. Hunt, M.S. "Synthesis of A Cascaded Network by Transfer Matrix Factorization", IRE (London) pp. 203-205, March 1967.

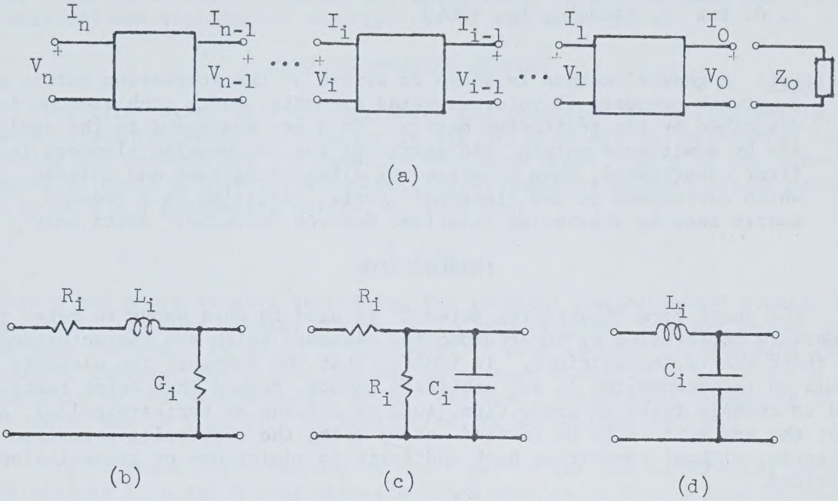

I_n
V_n
I_{n-1}
V_{n-1}
I_i
V_i
I_{i-1}
V_{i-1}
I_1
V_1
I_0
V_0
Z_0
(a)
R_i
L_i
G_i
(b)
R_i
R_i
C_i
(c)
L_i
C_i
(d)

Figure 1

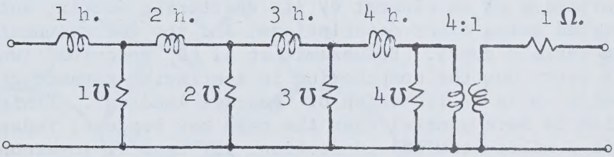

1 h.
2 h.
3 h.
4 h.
4:1
1 Ω.
1℧
2℧
3℧
4℧

Figure 2

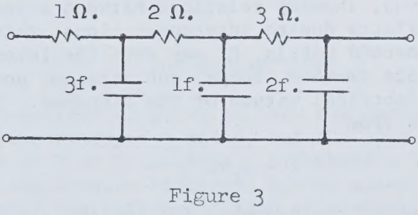

1 Ω.
2 Ω.
3 Ω.
3f.
1f.
2f.

Figure 3

ON THE ANALYSIS OF SCATTERING NETWORKS

JACOB SHEKEL
Jerrold Electronics Corporation
P. O. Box 37, Hatboro, Pa. 19040

Abstract: A general method is shown to arrive at the scattering matrix of a network composed of interconnected elements, where each element is described by its scattering matrix. In a way analogous to the analysis by admittance matrix, the matrix of the unconnected elements is first constructed, then condensed by eliminating rows and columns which correspond to the "internal" ports, resulting in a reduced matrix showing scattering relations between "external" ports only.

INTRODUCTION

The short term "Scattering Network" is used in this paper to refer to a network constructed by interconnecting elements which are characterized by their scattering matrices. It implies that the ports of the elements could be interconnected in any arbitrary manner, rather than being restricted to certain types of connection, such as cascade or series-parallel; and that the analysis is to be carried out by using the scattering formalism directly, without converting back and forth to admittance or transmission matrices.

There are many practical advantages to such an approach, in particular in broadband and high frequency networks, where many elements are most easily characterized -- and measured -- by scattering parameters, defined externally at the ports, without recourse to an equivalent network [1].

Numerous general computer programs have been described in the literature for the analysis of microwave circuits [2-4]. Most programs will accept a description of an element by its scattering matrix, but the computation is carried using other descriptions, and the interconnections are restricted to certain types. Bodharamik et al [5] described two general programs that carry out the computation in scattering parameters, and the network is built up by an iteration of "cascade loading". (This type of interconnection is more general than the term may suggest; indeed, since the "cascading" is a multi-terminal connection, any type of interconnection can be represented by this type of loading.)

When a network is analysed by impedance or admittance, there is a basic general approach which makes the analysis amenable to computer methods. It is possible to separate the description of the network into two parts, the elements and their interconnections. The elements are described by a branch matrix Y or Z, which is essentially diagonal (with off-diagonal entries for coupling between branches), showing relations between branch voltages and currents, but not the effects due to interconnections; this is Kron's "primitive network" [6]. A second matrix, C, may show the interconnections of branches into certain sets (meshes, loops, cut sets, or node incidence), without regard to the electrical nature of the branches. The complete network description results from

$$\overline{Y} = C_t Y C$$

which combines the nature of the branches (Y) and the topology of the network (C).

Abele [7] has described a method for calculating the scattering matrix for interconnected multipoles, where the interconnections are treated as constraints imposed on the matrix of unconnected multipoles. The purpose of this paper is to show a similar approach, which completely separates the electrical and topological descriptions of the network.

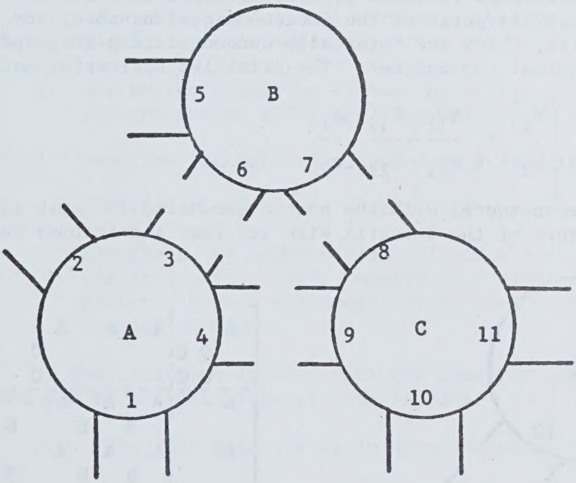
THE UNCONNECTED NETWORK

The first step, that of describing the primitive scattering matrix, is quite straightforward. Assume (as in Fig. 1) that all the multiports comprising the network are disconnected from each other. If $\underline{n}$ is the total number of ports of all the elements, the unconnected network can be described by an n x n scattering matrix, S, such that

$$b = Sa$$

where $\underline{a}$ and $\underline{b}$ are vectors describing the incident and reflected signals at the ports. In this matrix, any entry $S_{jk} = 0$ if $\underline{j}$ and $\underline{k}$ do not belong to the same element. In fact, if the ports of each element are assigned consecutive numbers, the scattering matrix of each m-port will be an m x m submatrix on the main diagonal of S. (However, a different numbering of the ports may be preferred to facilitate the steps that follow.)

Fig. 1 shows the structure of the S matrix for the unconnected network. The letters A, B and C only indicate which entries of S will have values representing element A, B or C. (All other entries of S are zero.)



$$\begin{bmatrix} A & A & A & A & & & & & & & \\ A & A & A & A & & & & & & & \\ A & A & A & A & & & & & & & \\ A & A & A & A & & & & & & & \\ & & & & B & B & B & & & & \\ & & & & B & B & B & & & & \\ & & & & B & B & B & & & & \\ & & & & & & & C & C & C & C \\ & & & & & & & C & C & C & C \\ & & & & & & & C & C & C & C \\ & & & & & & & C & C & C & C \end{bmatrix}$$

Fig. 1

INTERCONNECTIONS

The interconnections of ports to form the network introduce some constraints on the components of $\underline{a}$ and $\underline{b}$, but the type of constraint is different from those in Y or Z analysis. Kirchhoff's laws impose constraints between currents and between voltages, but no mixed constraints between voltage and current; the interconnection of scattering matrices introduces constraints between a's and b's. If a pair of ports $\underline{j}$ and $\underline{k}$ are connected together (whether they belong to the same element or not), a set of relationships between a_j, a_k, b_j and b_k is imposed.

Assume first that the incident and reflected signals at both ports are defined relative to the same characteristic impedance; then the interconnection of the two ports means that what is considered a reflected signal in one is the incident signal on the other, and vice versa; that is

$$\begin{bmatrix} b_j \\ b_k \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_j \\ a_k \end{bmatrix} \qquad (1)$$

In general, the signals at both ports may be defined relative to different impedances,

$$\begin{bmatrix} a_j \\ b_j \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & R_j \\ 1 & -R_j \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} \qquad \begin{bmatrix} a_k \\ b_k \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -R_k \\ 1 & R_k \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix}$$

(note that the same i that goes into the j-port comes out of the k-port, hence the difference in signs), then

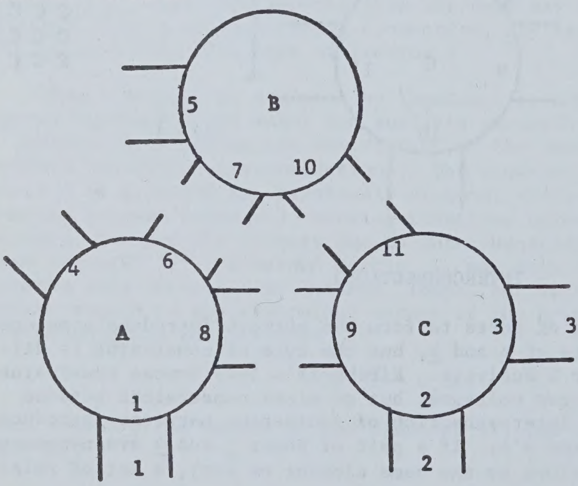
$$\begin{bmatrix} b_j \\ b_k \end{bmatrix} = \frac{1}{R_j+R_k} \begin{bmatrix} R_k-R_j & 2R_j \\ 2R_k & R_j-R_k \end{bmatrix} \begin{bmatrix} a_j \\ a_k \end{bmatrix} \qquad (2)$$

which reduces to the simpler relation above when $R_j = R_k$.

Let the ports be partitioned into two groups: group 1 of "external" ports, which are the accessible ports of the interconnected network, and group 2 of "internal" ports, which are internally connected in pairs, and are not available for external connections. The primitive scattering matrix can be partitioned into

$$\begin{bmatrix} b_1 \\ \hline b_2 \end{bmatrix} = \left[\begin{array}{c|c} S_{11} & S_{12} \\ \hline S_{21} & S_{22} \end{array}\right] \begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix} \qquad (3)$$

Fig. 2 shows the same network, with the ports renumbered for easy partitioning, and the structure of the S matrix with its rows and columns renumbered to match.



$$\left[\begin{array}{ccc|cccccccc}
A & & & A & & A & & A & & & \\
& C & C & & & & & & C & & C \\
& C & C & & & & & & C & & C \\
\hline
A & & & A & & A & & A & & & \\
& & & & B & & B & & & B & \\
A & & & A & & A & & A & & & \\
& & & & B & & B & & & B & \\
A & & & A & & A & & A & & & \\
& C & C & & & & & & C & & C \\
& & & & B & & B & & & B & \\
& C & C & & & & & & C & & C
\end{array}\right]$$

Fig. 2

The interconnections impose a constraint on b_2 and a_2, described by a Junction matrix, J

$$b_2 = Ja_2 \tag{4}$$

whose non-zero components correspond to pairs of interconnected ports. The Junction matrix combines the information about the interconnections, and the reference impedances at the involved ports, but is independent of the electrical nature of the interconnected multiports.

Substituting the constraint into the primitive relation,

$$\left[\begin{array}{c} b_1 \\ \hline 0 \end{array}\right] = \left[\begin{array}{c|c} S_{11} & S_{12} \\ \hline S_{21} & S_{22}-J \end{array}\right] \left[\begin{array}{c} a_1 \\ \hline a_2 \end{array}\right] \tag{5}$$

and the scattering matrix for the external ports can be formally computed as

$$b_1 = (S_{11}-S_{12}(S_{22}-J)^{-1}S_{21})a_1 \tag{6}$$

In practical computation, the solution need not necessarily be obtained by this formal method. It can be recognized that this result is in effect a reduction of the matrix by condensation [8], and any one of many well known numerical methods may be used, to yield results with adequate accuracy.

SUMMARY OF METHOD

The introduction of the Junction Matrix enables to set up a general algorithm for the solution of scattering networks, analogous to the steps of solving by admittance methods:

1. Set up the primitive scattering matrix, by grouping the matrices of all components along the diagonal of the matrix.
2. Describe the interconnections by a junction matrix.
3. Subtract the junction matrix from the primitive scattering matrix.
4. Eliminate, by condensation, all rows and columns corresponding to internal ports. This results in a scattering matrix for the external ports, which can now be used as a starting point for the problem on hand.

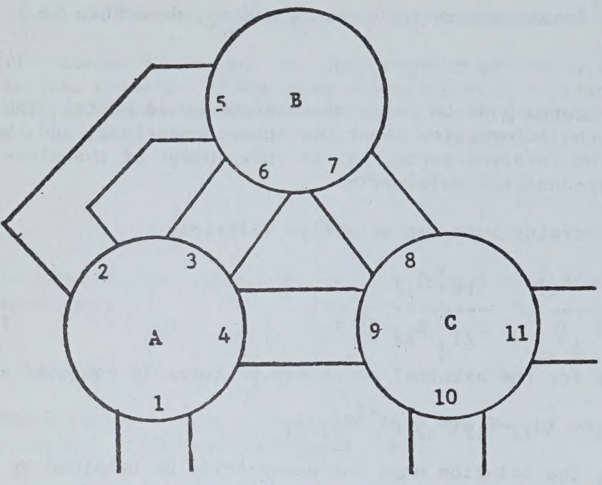
If the reference impedance is the same for all ports, steps 2 and 3 above can be combined into the simple rule:

2-3: For each pair j-k of interconnected ports, let

$$S_{jk} = S_{kj} = -1$$

Fig. 3 shows the application of the algorithm to the network of Fig. 1.

The "x" indicates the position of the "-1" terms; any row or column with an "x" is to be eliminated by condensation.



$$\begin{bmatrix}
A & A & A & A & & & & & & & \\
A & A & A & A & x & & & & & & \\
A & A & A & A & & x & & & & & \\
A & A & A & A & & & & & x & & \\
 & x & & & B & B & B & & & & \\
 & & x & & B & B & B & & & & \\
 & & & & B & B & B & x & & & \\
 & & & & & & x & C & C & C & C \\
 & & & x & & & & C & C & C & C \\
 & & & & & & & C & C & C & C \\
 & & & & & & & C & C & C & C
\end{bmatrix}$$

Fig. 3

MULTIPLE-PORT JUNCTIONS

The procedure, as described so far, only allows for pair-wise interconnections of ports. When a junction involves more than 2 ports, there may be different ways of making the interconnection; e.g., 3 ports can be interconnected in parallel or in series, and 6 ports can be made to form a bridge junction. Fig. 4(a), on the next page, shows an example of a scattering network with two multiple-port junctions.

It is possible to work out the constraints on the set of incident and reflected signals at such a junction and incorporate it into the junction matrix J. An equivalent result can be obtained if the junction itself -- whatever its complexity -- is considered to be another element of the network. In Fig. 4(b), the elements P and Q represent the multiple-port junctions of Fig. 4(a); if the scattering matrices of P and Q are included in the starting S, all the interconnections are reduced to simple port-pair connections, and the general procedure can be applied.

The scattering matrix of an n-port parallel junction can be derived from Fig. 5. Assuming all port impedances equal (normalized to 1), the input admittance at any one port, with all others terminated as in Fig. 5(b) is

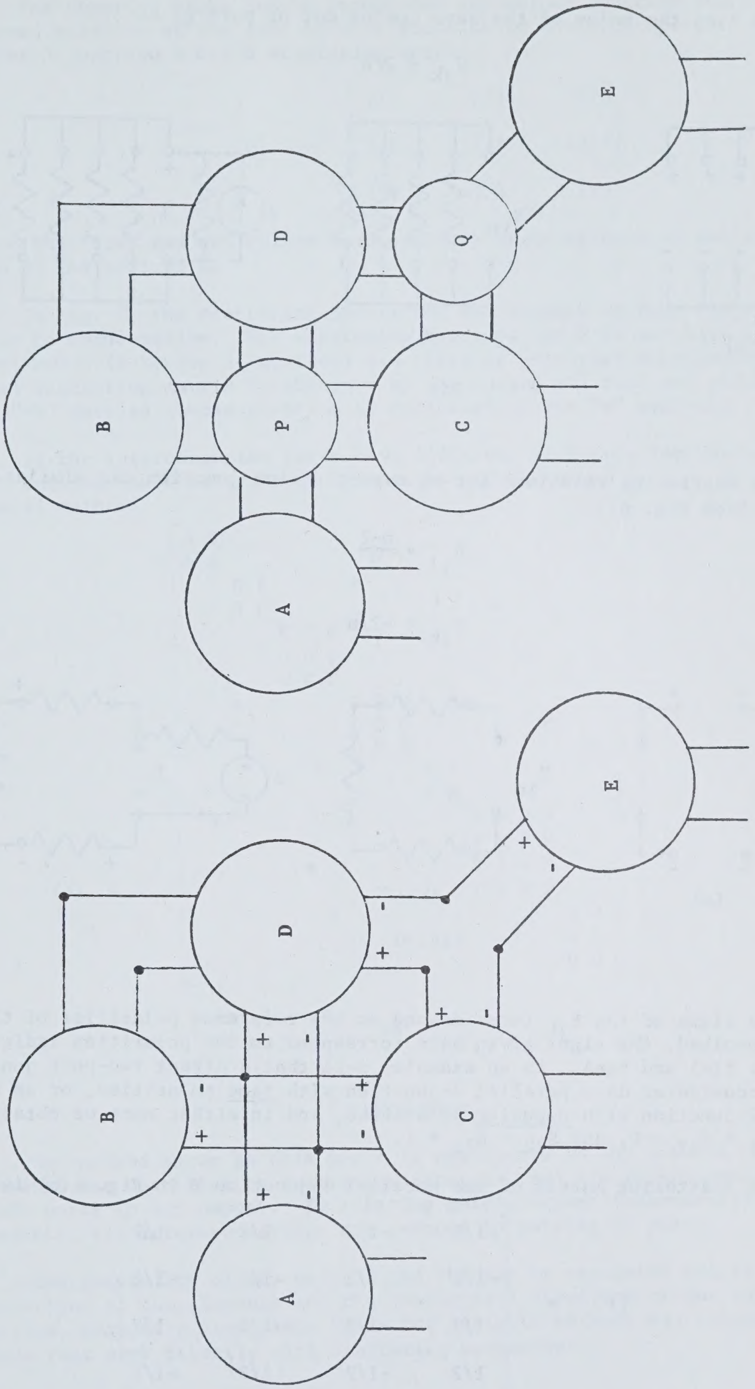
$$Y_{in} = n-1$$

therefore

$$S_{jj} = \frac{1-(n-1)}{1+(n-1)} = \frac{2-n}{n} \qquad (6a)$$

For the transfer ratio, assume a source of 2 amperes with a 1 ohm internal admittance applied to port j, as in Fig. 4(c), providing a unit incident wave. The current into a termination at port $\underline{k}$ is

$$i_k = 2/n$$



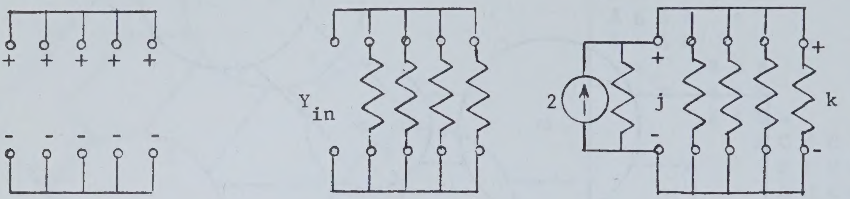
(a)

(b)

FIG. 4

which is also the value of the wave coming out of port k, so

$$S_{jk} = 2/n \tag{6b}$$

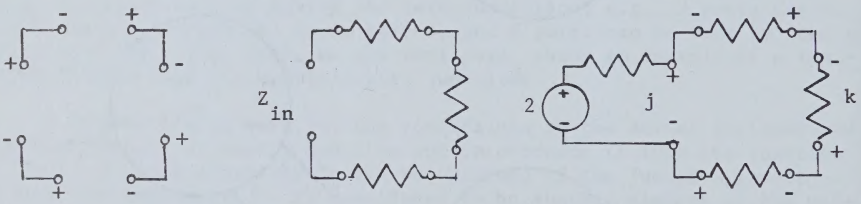


(a) (b) (c)

Fig. 5

The scattering relations for an n-port series junction can similarly be derived from Fig. 6:

$$\begin{aligned} S_{jj} &= \frac{n-2}{n} \\ S_{jk} &= -2/n \end{aligned} \tag{7}$$



(a) (b) (c)

Fig. 6

The signs of the S_{jk} terms depend on the reference polarities of the ports involved, the signs given here correspond to the polarities indicated in Figs. 5(a) and 6(a). As an example, note that a direct two-port junction can be considered as a parallel 2-junction with same polarities, or as a series 2-junction with opposite polarities, and in either case we obtain that $S_{11} = S_{22} = 0$, and $S_{12} = S_{21} = 1$.

The scattering matrix of the parallel 4-junction P in Fig. 4(b) is

$$[P] = \begin{bmatrix} -1/2 & -1/2 & 1/2 & 1/2 \\ -1/2 & -1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \end{bmatrix}$$

The negative signs in the second row and column indicate that the assumed polarity at the port of B is opposite to the other ports. The series 3-junction Q has a scattering matrix

$$[Q] = \begin{bmatrix} 1/3 & 2/3 & 2/3 \\ 2/3 & 1/3 & -2/3 \\ 2/3 & -2/3 & 1/3 \end{bmatrix}$$

where the first row and column have positive signs because of the orientation of the port of C.

In Fig. 7, the scattering matrix for the network of Fig. 4(b) is shown, prior to condensation. The scattering matrices for P (a parallel 4-junction) and Q (a series 3-junction) are shown as principal sub-matrices. The final scattering matrix is obtained by condensing all rows and columns with the "-1" entries (whose position is indicated by the "x" symbol).

If the interconnected ports have different reference impedances, the "-1" entries would be replaced by entries such as given by (2), as in the general method.

$$\left[\begin{array}{ccccccccccccccccccc}
A & A & & & & & & & & & & & & & & & & & \\
A & A & & & & & & & & & & & x & & & & & & \\
 & & B & B & & & & x & & & & & & & & & & & \\
 & & B & B & & & & & & & & & & x & & & & & \\
 & & & & C & C & C & & & & & & & & & & & & \\
 & & & & C & C & C & & & & & & & & x & & & & \\
 & & & & C & C & C & & & & & & & & & & x & & \\
 & & x & & & & & D & D & D & & & & & & & & & \\
 & & & & & & & D & D & D & & & & & & x & & & \\
 & & & & & & & D & D & D & & & & & & & & x & \\
 & & & & & & & & & & E & E & & & & & & & \\
 & & & & & & & & & & E & E & & & & & & & x \\
 & x & & & & & & & & & & & P & P & P & P & & & \\
 & & & x & & & & & & & & & P & P & P & P & & & \\
 & & & & & x & & & & & & & P & P & P & P & & & \\
 & & & & & & & & x & & & & P & P & P & P & & & \\
 & & & & & & x & & & & & & & & & & Q & Q & Q \\
 & & & & & & & & & x & & & & & & & Q & Q & Q \\
 & & & & & & & & & & & x & & & & & Q & Q & Q
\end{array}\right]$$

Fig. 7

CONCLUSION

The method shown in this paper is applicable to any network whose elements are described by scattering matrices, and are interconnected at their ports in any manner. By treating multiple-port interconnections as elements, all interconnections are reduced to pairing of ports.

The generality of the method, and the way it separates the electrical properties of the elements and the topological structure of the interconnection, make it a convenient basis for analysis methods and computer programs that work directly with scattering parameters.

REFERENCES

1. "S-parameters...circuit analysis and design", Hewlett-Packard application note 95, Sept. 1968.

2. M. N. Parker, "DIPNET: a general distributed parameter network analysis program", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-17, Aug. 1969, pp. 495-505.

3. P. L. Green, "General purpose programs for the frequency domain analysis of microwave circuits", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-17, Aug. 1969, pp. 506-514.

4. T. N. Trick and J. Vlach, "Computer-aided design of broadband amplifiers with complex loads", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-18, Sept. 1970, pp. 541-547.

5. P. Bodharamik, L. Besser and R. Newcomb, "Two scattering matrix programs for active circuit analysis", IEEE Trans. on Circuit Theory, vol. CT-18, Nov. 1971, pp. 610-619.

6. G. Kron, Tensor Analysis of Networks, Wiley, 1939.

7. Th.-A. Abele, "Über die Streumatrix allgemein zusammengeschalteter Mehrpole", Arch. Elek. Übertragung, vol. 14, pp. 262-268, 1960.

8. J. Shekel, "On transformation groups and equivalent networks", in book "Aspects of Network and System Theory", edited by R. E. Kalman and N. DeClaris; Holt, Rinehart and Winston, 1971.

ON A DIGITAL OSCILLATOR

DEBASIS MITRA and H. D. MONTOGMERY
Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

This paper analyzes some of the basic problems associated with the design of a robust oscillator. The device uses a second-order digital filter section and a simple nonlinearity. The design parameters of the system are the two multiplier coefficients of the second-order filter section and two additional parameters associated with the nonlinearity. Conditions on the design parameters are derived which, when satisfied, guarantee the existence of the desired oscillatory behavior.

I. Introduction

The communication systems of the future will probably use digital processing extensively. Some of these systems will require simple and reliable discrete oscillators for their operations. The results of this paper concern a nonlinear discrete oscillator which uses a second-order digital filter section and a simple nonlinearity.

One of the uses foreseen for this digital oscillator is as a frequency shift keying (FSK) digital modulator. Briefly, the ideal FSK modulator oscillates at two widely separated frequencies in correspondence with a binary process. Under certain conditions it appears possible to get adequate frequency shift keying from the device under consideration by appropriately switching a single parameter (β_1 in Figure 1). However, in order to establish the feasibility of the device as a FSK modulator it is certainly necessary to first establish, as is done here, the conditions under which reasonable oscillatory behavior can be guaranteed in the absence of shift keying.

The device is shown in Figure 1. The nonlinearity is given thus:

$$
\begin{aligned}
F(\sigma) &= \sigma & &\text{if } |\sigma| \leq \delta \\
&= (1-\alpha)\sigma+\alpha\delta & &\text{if } \sigma \geq \delta \\
&= (1-\alpha)\sigma-\alpha\delta & &\text{if } \sigma \leq -\delta ,
\end{aligned}
$$

where α and δ are positive constants and also $\alpha < 1$. The design parameters of the device are therefore β_1, β_2, α and δ.

The paper gives conditions for there to exist a region -- the oscillation region -- in the state space of the device with certain associated properties and characterized thus:

$$\text{oscillation region:} \quad a < \|z\|_{Q_1} \text{ and } \|z\|_{Q_2} < b ,$$

where z is the generic state vector, $\|\cdot\|_{Q_i}$ denotes a norm, and a and b are positive constants. The first property associated with the oscillation region is that the state vector is forced into the oscillation region in a finite number of transitions for all initial states which are not elements of a small neighborhood of the null state. Secondly, transitions from the oscillation region to its complement are impossible. Thirdly, there do not exist equilibirum points in the oscillation region. The paper also gives lower and upper bounds on the root-mean-square (r.m.s.) value of the oscillations, and an upper bound on the amplitude of the oscillations.

The results of extensive simulations concerning the device as an oscillator are given in reference 1. This paper makes extensive use of Liapunov techniques and the basic variational properties of convex functions; appropriate references are 2 and 3 respectively.

2. Preliminaries and Notation

2.1 The equations For the ideal filter, in the absence of quantization and other noise effects, the equations are

$$x_{n+1} = F(\beta_2 x_{n-1} + \beta_1 x_n) \qquad n = 0,1,2,\ldots \; . \tag{1}$$

$$= \begin{cases} \beta_2 x_{n-1} + \beta_1 x_n & \text{if } |\beta_2 x_{n-1} + \beta_1 x_n| \le \delta \\ (1-\alpha)(\beta_2 x_{n-1} + \beta_1 x_n) + \alpha\delta & \text{if } \beta_2 x_{n-1} + \beta_1 x_n \ge \delta \\ (1-\alpha)(\beta_2 x_{n-1} + \beta_1 x_n) - \alpha\delta & \text{if } \beta_2 x_{n-1} + \beta_1 x_n \le -\delta \end{cases} \tag{2}$$

In reality however the equations governing the sequence $\{x_n\}$ will be the following

$$x_{n+1} = F(\beta_2 x_{n-1} + \beta_1 x_n + e'_n) + e''_n \, , \qquad n = 0,1,2,\ldots \; . \tag{3}$$

Hence

$$x_{n+1} = F(\beta_2 x_{n-1} + \beta_1 x_n) + e_n \qquad n = 0,1,2,\ldots \; , \tag{4}$$

where e_n is defined by (3) and (4). It will be assumed throughout that $|e_n|$ is uniformly bounded by Δ, i.e., $|e_n| \le \Delta$, $n = 0,1,2,\ldots$. Since $\{e_n\}$ represents mainly the effect of quantization, Δ can be made small by using a sufficient number of bits in the digital representation of signals.

2.2 The state equations The state at the n^{th} instant is denoted by z_n, $z_n \in R^2$. Also,

$$z_n = \begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix}. \tag{5}$$

The generic vector in R^2 is denoted by z. Two forms of the equations governing the state transitions will be used. The first form is the following

$$\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \beta_2 & \beta_1 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u_n \tag{6}$$

where

$$u_n = G(\beta_2 x_{n-1} + \beta_1 x_n) + e_n \, , \tag{7}$$

and

$$G(\sigma) = \begin{cases} -\alpha(\sigma-\delta) & \text{if } \sigma > \delta \\ 0 & \text{if } |\sigma| \le \delta \\ -\alpha(\sigma+\delta) & \text{if } \sigma < -\delta \, . \end{cases} \tag{8}$$

Equation (6) in matrix form is

$$z_{n+1} = A_1 z_n + b u_n \tag{6'}$$

where the definitions of A_1 and b are self-explanatory.

The second useful form of the state transition equations is the following

$$\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ (1-\alpha)\beta_2 & (1-\alpha)\beta_1 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v_n \tag{9}$$

where $\quad v_n = H(\beta_2 x_{n-1} + \beta_1 x_n) + e_n \qquad (10)$

and

$$H(\sigma) = \begin{cases} \alpha\delta & \text{if} \quad \sigma \geq \delta \\ \alpha\sigma & \text{if} \quad |\sigma| \leq \delta \\ -\alpha\delta & \text{if} \quad \sigma \leq -\delta \end{cases} \quad . \tag{11}$$

Equation (9) in matrix form is

$$z_{n+1} = A_2 z_n + b v_n \tag{9'}$$

where the definition of A_2 is self-explanatory.

The vector $(\beta_2 \; \beta_1)$ is denoted by β^t in the sequel.

2.3 Associated linear systems It will be assumed that both eigenvalues of A_1 lie outside the unit circle and that both eigenvalues of A_2 lie inside the unit circle. It is easy to see that the assumptions are necessary to ensure that the state space response of the system does not collapse or blow up. (Some additional observations on the necessity of the above assumptions are contained in Section 7.) It may be verified that the above assumptions on the eigenvalues of A_1 and A_2 are equivalent to the following conditions on β_1 and β_2:

either, $\quad -\frac{1}{1-\alpha} < \beta_2 < -1 \quad$ and $\quad |\beta_1| + \beta_2 < 1 \qquad (12)$

or, $\quad |\beta_1| + \beta_2 < \frac{1}{1-\alpha} \quad$ and $\quad -|\beta_1| + \beta_2 > 1 \qquad (13)$

In Figure 2 the region in the $\beta_1 - \beta_2$ plane where the above conditions are satisfied is cross-hatched.

As is well known [2] the condition on A_2 is also equivalent to the following: for every given symmetric positive definite matrix P there exists a symmetric positive definite matrix Q such that*

$$Q - A_2^t Q A_2 = P \tag{14}$$

From the above statement it can also be deduced that if both eigenvalues of A_1 are outside the unit circle then for every given symmetric positive definite matrix P there exists a symmetric positive definite matrix Q such that

$$A_1^t Q A_1 - Q = P \tag{15}$$

---

* X^t denotes the transpose of X .

2.4 Notation The Euclidean norm of a vector $z \in R^2$ is denoted by $||z|| = \sqrt{z_1^2 + z_2^2}$. A generalized norm is $||z||_Q = \sqrt{z^tQz}$. The eigenvalues of Q are denoted by $\{\lambda_i(Q)\}$; $\lambda_{min}(Q)$ and $\lambda_{max}(Q)$ respectively denote the smallest and largest eigenvalues of Q.

3. Instability Region

A region in the state space of the nonlinear filter is defined with the property that if initially the state vector is an element of this region then after only a finite number of transitions the state vector is forced to the exterior of the region. It is appropriate to call this region the instability region since a norm of the state vector is monotonic, increasing with time during the period the state vectors are elements of the region. One of the effects of quantization is that a small neighborhood of the origin in state space is 'seen' as the null i.e., quiescent state and this effect is also analyzed.

Consider equation (6′) and suppose that $|\beta^t z_n| \leq \delta$. Now $G(\beta^t z_n) = 0$ and $u_n = e_n$. Let P and Q be two symmetric positive definite matrices such that $A_1^tQA_1 - Q = P$. Since it is easily seen that

$$z_{n+1}^t Q z_{n+1} - z_n^t Q z_n$$

$$\geq \lambda_{min}(Q^{-1}p)||z_n||_Q^2 - 2\Delta||A_1^tQb||_{Q^{-1}} \, ||z_n||_Q$$

it follows that

$$z_{n+1}^t Q z_{n+1} > z_n^t Q z_n \quad \text{when} \quad z_n^t Q z_n > \left\{ \frac{2\Delta||A_1^tQb||_{Q^{-1}}}{\lambda_{min}(Q^{-1}P)} \right\}^2 .$$

It has been shown that if z_0 is not an element of a neighborhood N(0) of the origin,

$$N(0) \triangleq \left[z \,\middle|\, z^tQz \leq \left\{ \frac{2\Delta||A_1^tQb||_{Q^{-1}}}{\lambda_{min}(Q^{-1}P)} \right\}^2 \right], \tag{16}$$

then in the region $|\beta^t z| \leq \delta$, a norm of the state vectors, $||z_n||_Q$, increases monotonically. Observe that this neighborhood of the origin can be made small by using sufficient number of bits in the digital representation.

3.1 Maximum Instability Region What is the maximum value of the norm $||z_n||_Q$ for which the aforementioned monoticity property holds? The geometrical problem is to obtain the maximum value of K^2 such that the ellipse: $z^tQz = K^2$ is contained entirely in the region $|\beta^t z| \leq \delta$. From the geometrical interpretation of K^2 it is clear that

$$K^2 = \min_{|\beta^t z| = \delta} z^tQz \tag{17}$$

The method of Lagrange multipliers yields the following unique solution to the variational problem in (17):

$$K^2 = \frac{\delta^2}{\beta^t Q^{-1} \beta} \tag{18}$$

Two key properties of the solution are

(a) $|\beta^t z| \geq \delta$ implies $z^t Q z \geq \delta^2/\beta^t Q^{-1}\beta,$ (19)

and

(b) $z^t Q z \leq \delta^2/\beta^t Q^{-1}\beta$ implies $|\beta^t z| \leq \delta$. (20)

Property (b) is of immediate interest and property (a) is used in Section 4.

3.2 Geometric Growth For some $\varepsilon > 0$ consider the following neighborhood of the origin:

$$N_\varepsilon(0) = \left\{ z \,\middle|\, z^t Q z \leq \left(\frac{2\Delta\|A_1^t Qb\|_{Q^{-1}}}{\lambda_{\min}(Q^{-1}P)} \right)^2 + \varepsilon \right\}$$

The instability region is defined thus:

$$\left\{ z \,\middle|\, z \notin N_\varepsilon(0) \text{ and } z^t Q z \leq \frac{\delta^2}{\beta^t Q^{-1}\beta} \right\}$$

It may reasonably be assumed that the instability region is not null; in most cases of practical interest it is a substantial region because Δ is a comparatively small number.

The following assertion can be proved to be true: if z_0 is an element of the instability region then there exists a finite N such that $z_N^t Q z_N > \dfrac{\delta^2}{\beta^t Q^{-1}\beta}$. In fact, in the instability region, growth is geometric in the sense that there exists a positive constant ρ such that

$$z_n^t Q z_n > (1+\rho)^n z_0^t Q z_0.$$

when the states z_i, $i = 0,1,\ldots,n$ are elements of the instability region.

4. Transitions to the Instability Region from Its Exterior are Impossible

In this section we are concerned with the possibility of occurrence of a phenomenon which is peculiar to discrete systems since such an event can be immediately discounted in continuous systems purely from the continuity of trajectories in state space. In the analysis of the device under investigation we are, at this stage, left with the possibility of a transition from the exterior to the interior of the instability region at some instant. For the existence of reasonable oscillatory behavior of the nonlinear filter it is necessary to show that such transitions are impossible and this is done in this section. It should be noted however that this is not fortuitous but depends on the proper choice of the instability region.

It is proved below that if $\Delta < \delta$ then

$$z_n^t Q z_n \geq \delta^2/(\beta^t Q^{-1}\beta) \quad \text{implies} \quad z_{n+1}^t Q z_{n+1} \geq \delta^2/(\beta^t Q^{-1}\beta)$$

The proof is in four parts.

Case (i): $|\beta^t z_n| \leq \delta$.

From the hypothesis and the definition of the instability region

$$z_{n+1}^t Q z_{n+1} > z_n^t Q z_n .$$

Case (ii): $|\beta^t z_n| \geq \delta$ and $|\beta^t z_{n+1}| \geq \delta$.

From property (a) of the boundary of the unstability region, equation (19),

$$|\beta^t z_{n+1}| \geq \delta \quad \text{implies} \quad z^t_{n+1} Q z_{n+1} \geq \delta^2/(\beta^t Q^{-1}\beta) .$$

Case (iii): $\beta^t z_n \geq \delta$ and $|\beta^t z_{n+1}| \leq \delta$.

Since $\beta^t z_n \geq \delta$, it follows from (9′) that

$$z_{n+1} = A_2 z_n + b v_n \quad \text{where} \quad v_n = \alpha\delta + e_n .$$

The condition $|\beta^t z_{n+1}| \leq \delta$ is therefore equivalent to

$$-\delta - \beta^t b v_n \leq \beta^t A_2 z_n \leq \delta - \beta^t b v_n$$

Case (iii) therefore considers the event that z_n is an element of a closed region $S \subset R^2$ and the boundary of S consists of linear segments. Now,

$$z^t_{n+1} Q z_{n+1} = z^t_n A^t_2 Q A_2 z_n + 2 v_n b^t Q b + v^2_n b^t Q b$$

For fixed v_n, consider the above expression as a function of z_n and let $f(z_n) \triangleq z^t_{n+1} Q z_{n+1}$. Note that $f(z_n)$ is convex. The minimum of $f(z_n)$ for $z_n \in S$ occurs either at the boundary of S or at a stationary point. At the boundary of S, either $\beta^t z_n = \delta$ or $[\beta^t z_n \geq \delta$ and $|\beta^t z_{n+1}| = \delta]$. If $\beta^t z_n = \delta$ then from Case (i) the implication holds i.e. $f(z_n) \geq \delta^2/(\beta^t Q^{-1}\beta)$. If $\beta^t z_n \geq \delta$ and $|\beta^t z_{n+1}| = \delta$ then from Case (ii) the implication holds.

Now consider the stationary point. If $\hat{z}$ is a stationary point of $f(z_n)$ then

$$A^t_2 Q A_2 \hat{z} + v_n A^t_2 Q b = 0$$

i.e.

$$\hat{z} = -v_n A^{-1}_2 b .$$

The reader may verify that

$$\beta^t \hat{z} = -v_n \beta^t A^{-1}_2 b = \frac{-v_n}{1-\alpha} = \frac{-(\alpha\delta + e_n)}{1-\alpha} .$$

Hence, if $|e_n| < \delta$ then $\beta^t \hat{z} < \delta$ and the stationary point $\hat{z}$ is not an element of S.

It has been shown that if $\Delta < \delta$ then the minimum of $f(z_n)$ occurs at the boundary of S and at the boundary $f(z_n) \geq \delta^2/(\beta^t Q^{-1}\beta)$.

Case (iv): $\beta^t z_n \leq -\delta$ and $|\beta^t z_{n+1}| \leq \delta$.

The proof for this case is similar to the proof for Case (iii).

5. The Stability Region

In the state space of the nonlinear filter there exists a region of the form $\{z | z^t Q z \geq K^2\}$, Q symmetric and positive definite, such that if z_0 is an element of this region then after only a finite number of transitions, say N, the state vector lies in the complement of the region i.e. $z^t_N Q z_N < K^2$. The region is called the stability region since, during the period that the state vectors are elements of the region, a norm of the state vectors decreases monotonically with time.

The state transition equation (9′) is used throughout this section. Let P and Q be two symmetric positive definite matrices such that $Q - A_2^t Q A_2 = P$. Then

$$z_n^t Q z_n - z_{n+1}^t Q z_{n+1} = z_n^t P z_n - 2v_n b^t Q A_2 z_n - v_n^2 b^t Q b \ . \tag{21}$$

Observe that $|v_n| \leq \alpha\delta + \Delta$. Let $v \overset{\Delta}{=} \alpha\delta + \Delta$. Now, to define the stability region it is easy to see that it is necessary to find K^2 where K^2 is the smallest number such that

$$z^t Q z \geq K^2 \quad \text{implies} \quad z^t P z - 2v\, b^t Q A_2 z - v^2 b^t Q b \geq 0.$$

To obtain the quantity K^2 as defined above it is necessary to solve the following variational problem.

$$K^2 = \max_{z^t P z - 2v\, b^t Q A_2 z - v^2 b^t Q b = 0} z^t Q z \ . \tag{22}$$

The variational problem in (22) becomes clearer if certain transformations are made -- the transformations may be deduced from the following form.

$$K^2 = \max_{y^t \Lambda y = d^2} (y-\eta)^t (y-\eta) \tag{23}$$

where Λ is the diagonal matrix with the eigenvalues of $Q^{-1}P$ as diagonal elements, $d^2 = v^2 b^t (Q A_2 P^{-1} A_2^t Q + Q) b$, and $\eta = -vU^t Q^{1/2} P^{-1} A_2^t Q b$ where U is the orthogonal matrix with eigenvectors of $Q^{-1/2} P Q^{-1/2}$ as columns.

The variational problem now has a succinct geometrical statement: K is the smallest radius of those circles with center η that contain the ellipse $y^t \Lambda y = d^2$. The exact solution of the problem is not trivially obtained. The method of Lagrange multipliers yields a solution in terms of a root of a quartic. This is left as an exercise, and instead we obtain a bound $C^2 \geq K^2$. The bound C^2 is obtained by first containing the ellipse in a circle with the origin as center and this circle is next contained in another circle with η as center.

On the basis of the above observations, it is asserted that

$$C^2 \overset{\Delta}{=} \left[\|\eta\| + \frac{d}{\sqrt{\lambda_{\min}(Q^{-1}P)}} \right]^2 \tag{24}$$

$$\geq K^2 \tag{25}$$

where K^2 is as defined in (23). We omit the proof of (25).

It may be observed that the bound C^2 reduces to K^2 in two cases: (i) η is along the major axis of the ellipse $y^t \Lambda y = d^2$, and (ii) the ellipse is also a circle i.e. $\Lambda_{11} = \Lambda_{22}$. The expression for C in (24) is equivalent to the following expression stated in terms of the original variables:

$$\begin{aligned} C &= \|\eta\| + d/\sqrt{\lambda_{\min}(Q^{-1}P)} \\ &= (\alpha\delta + \Delta) \left[\|P^{-1} A_2^t Q b\|_Q + \sqrt{\lambda_{\max}(P^{-1}Q)(\|A_2^t Q b\|_{P-1}^2 + \|b\|_Q^2)} \right] \end{aligned} \tag{26}$$

For some $\varepsilon > 0$ let $\{z \mid z^t Qz > C^2 + \varepsilon\}$ denote the stability region. It has been shown that if z_n is an element of the stability region then $z_n^t Qz_n > z_{n+1}^t Qz_{n+1}$. It can be shown that there exists a positive constant $\rho < 1$ with the property that if z_i, $i = 0,1,2,\ldots,n$ are elements of the stability region, then

$$\|z_{n+1}\|_Q \leq \rho^{n+1} \|z_0\|_Q \ .$$

6. Transitions to the Stability Region from Its Complement are Impossible

It is required to show that

$$z_n^t Qz_n \leq C^2 \qquad \text{implies} \qquad z_{n+1}^t Qz_{n+1} \leq C^2 \ . \tag{27}$$

There are two cases to consider. In the following $v = \alpha\delta + \Delta$.

Case (i): $$(z_n - vP^{-1}A_2^t Qb)^t P(z_n - vP^{-1}A_2^t Qb) > v^2 b^t (QA_2 P^{-1} A_2^t Q + Q)b \tag{28}$$

and $$(z_n + vP^{-1}A_2^t Qb)^t P(z_n + vP^{-1}A_2^t Qb) > v^2 b^t (QA_2 P^{-1} A_2^t Q + Q)b \ . \tag{29}$$

Case (ii): Either (28) or (29) is violated.

Let $S \triangleq \{z_n \mid \text{either (28) or (29) is violated}\}$; S is a closed set. Equations (28) and (29) imply that $z_n^t Qz_n > z_{n+1}^t Qz_{n+1}$ and therefore the implication in (27) holds for Case (i).

From the preceding section,

$$z_{n+1}^t Qz_{n+1} \leq z_n^t A_2^t QA_2 z_n + 2v|b^t QA_2 z_n| + v^2 b^t Qb \ . \tag{30}$$

Let $f(z_n)$ denote the expression on the right hand side of (30). Since $f(z_n)$ is convex, the maximum value of $f(z_n)$ for $z_n \in S$ is attained at the boundary of S. But if z_n is an element on the boundary of S, $f(z_n) = z_n^t Qz_n \leq C^2$. The proof is complete.

7. Equilibrium Points

To guarantee oscillatory behavior of the nonlinear filter it is necessary to prohibit the existence of equilibrium points i.e., states such as z_n where $z_n = z_{n+1}$, since otherwise $z_i = z_n$ for all $i > n$. It can be deduced from what follows that if ideal conditions existed i.e., quantization effects etc. are absent and if all the eigenvalues of the matrix A_2 (Sec. 2.3) lie inside the unit circle then a necessary and sufficient condition for there not to exist any equilibrium points is: $1 - (\beta_2 + \beta_1) > 0$. This condition by itself is a new limitation on the parameters. If however quantization effects are considered then conditions of the form $1 - (\beta_1 + \beta_2) \geq \varepsilon > 0$ are necessary.

If z_n is an equilibrium point then $z_n = z_{n+1}$ or equivalently $x_{n-1} = x_n = x_{n+1}$.

Case (i): $|\beta^t z_n| \leq \delta$. An equilibrium point z_n, $z_n \neq 0$, exists if and only if $|(\beta_1 + \beta_2)x_n| \leq \delta$ where

$$x_n = \frac{e_n}{1-(\beta_2+\beta_1)} \ .$$

In connection with the above equations note that $|e_n| \leq \Delta$. An equilibrium point in N(0), a neighborhood of the origin defined in (16), may reasonably be considered a trivial equilibrium point as it is of little practical interest. From the definition of the instability region, an equilibrium point cannot exist in the instability region. Finally, it can be shown that there exists no equilibrium points in the remaining part of the region being considered in this case for the same conditions under which the instability region was defined.

Case (ii): $\beta^t z_n > \delta$. An equilibrium point z_n exists in this region if and only if,

$$x_n(\beta_2+\beta_1) \geq \delta$$

and

$$x_n[1-(1-\alpha)(\beta_2+\beta_1)] = \alpha\delta + e_n$$

Since both eigenvalues of the matrix A_2 lie inside the unit circle it is known that

$$1 - (1-\alpha)(\beta_2+\beta_1) > 0 \qquad \text{and} \qquad |\beta_2+\beta_1| < 3/(1-\alpha) \ .$$

With the help of these inequalities it can be shown that there exists no equilibrium point z_n, $\beta^t z_n \geq \delta$, if

$$1 - (\beta_1+\beta_2) > \frac{3\Delta}{\delta(1-\alpha)} \ . \tag{31}$$

Case (iii): $\beta^t z_n \leq -\delta$. In this case the fact that both eigenvalues of A_2 lie inside the unit circle is enough to show that no equilibrium point exists in the region.

This concludes the consideration of stationary points. Besides stationary points, it is also undesirable to have the following type of trivial oscillations: $x_{n-1} = -x_n$, $n = 1,2,\ldots$. The reader may verify that when (β_1, β_2) is not an element of the region which is symmetrical about the β_2-axis to the cross-hatched region in Figure 3, then it is impossible for the device to exhibit the above form of oscillations.

In conclusion, we observe that this section has derived conditions on the parameters which are not implied by those made initially (Section 2, Figure 2).

8. Limit Cycles and Bounds on R.M.S. Values and Amplitude

We have established that if the parameters of the nonlinear filter satisfy certain conditions and the initial state vector z_0 is not an element of $N_\varepsilon(0)$ then the state vector of the filter is forced in a finite number of transitions to a region devoid of stationary points and furthermore the state vector is thereafter confined to this region. It is appropriate to call this region the oscillation region. Since the digital filter is a finite state machine the existence of limit cycles in the oscillation region is guaranteed.

8.1 Bounds on r.m.s. values

The bounds on the oscillation region that have already been obtaineddirectly yield bounds on the r.m.s. values of the limit cycles. We first consider an upper bound. Recall that from the assumptions made on the eigenvalues of the matrix A_2 there exists, for any symmetric positive definite matrix P, a positive definite Q such that

$$Q - A_2^t Q A_2 = P \tag{14}$$

and for any such pair (P, Q), a constant C is defined in (26).

From the equations derived for the outer boundary of the oscillation region it can be shown that, if $x_1, x_2, \ldots, x_N$ are the elements of the observed limit cycle then,

$$\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right)^{1/2} \leq C/\sqrt{2\,\lambda_{\min}(Q)} \; .$$

Similarly, from the equations derived for the inner boundary of the oscillation region, it can be shown that

$$\delta/\sqrt{2\,\lambda_{\max}(Q)(\beta^t Q^{-1}\beta)} \leq \left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right)^{1/2}$$

where, now, $A_1^t Q A_1 - Q = P$.

and P is any symmetric positive definite matrix.

8.2 Bound on the amplitude. In the limit cycle oscillation let $x_1, x_2, \ldots, x_N$ respectively denote the values of the scalar output variables in the order that it is observed. It has been shown that with

$$z_1 = \begin{pmatrix} x_N \\ x_1 \end{pmatrix} \text{ and } \quad z_n = \begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix}, \quad n = 2,3,\ldots,N ,$$

the bound $z_n^t Q z_n \leq C^2$ holds. This bound and the special structure of the vectors z_n, $n = 1,2,\ldots,N$, are exploited to obtain an expression which yields by simple calculations an upper bound for the amplitude of the limit cycle.

Consider the following variational problem:

$$\min_{1\leq i\leq 2} \quad \max_{z^t Q z \leq C^2} \quad (e_i^t z)^2 \triangleq a^2 \tag{32}$$

where $e_i (i=1,2)$ is the vector with the i^{th} element equal to unity and the remaining element equal to zero. Clearly,

$$|x_n| \leq a \qquad n = 1,2,\ldots,N$$

and, also, a is the best amplitude bound that can be obtained from the quadratic bound.

The solution to the variational problem in (32) is obtained by the method of Lagrange multipliers. The reader may verify that

$$a^2 = \frac{C^2}{\det Q}\,\mathrm{Min}(Q_{11}, Q_{22})$$

We observe in conclusion that the methods utilized for deriving the quantity C and, subsequently, the amplitude bound a, can be applied in contexts more general than that considered in this paper; also, to the best of our knowledge, the bound and its derivation is new.

9. An Example

An example with the following choice of parameters is considered.

$$\beta_1 = 1.22, \ \beta_2 = -1.04, \ \alpha = 0.25, \ \delta = 0.75 \ .$$

The two matrices P occurring in equations (14) and (15) were chosen to be $\begin{bmatrix} 1.5 & -0.8 \\ -0.8 & 1.0 \end{bmatrix}$ and $\begin{bmatrix} 10/3 & -1/3 \\ -1/3 & 1 \end{bmatrix}$ respectively. The resulting boundaries of the oscillation region in state space are shown in Figure 4 by the dotted ellipses. Points on the limit cycle are indicated by crosses.

Acknowledgments

We benefited from discussions with J. F. Kaiser and H. S. McDonald who proposed the problem. We are very grateful to J. F. Kaiser for also assisting us in the computations.

References

1. H. D. Montgomery, "A Nonlinear Digital Oscillator", S. M. Thesis, Dept. of Elec. Eng., Massachusetts Institute of Technology, Cambridge, Mass., 1972.

2. J. L. Willems, "Stability Theory of Dynamical Systems", John Wiley, New York, 1970.

3. G. Hadley, "Nonlinear and Dynamic Programming", Addison-Wesley, Reading, Mass., 1964.

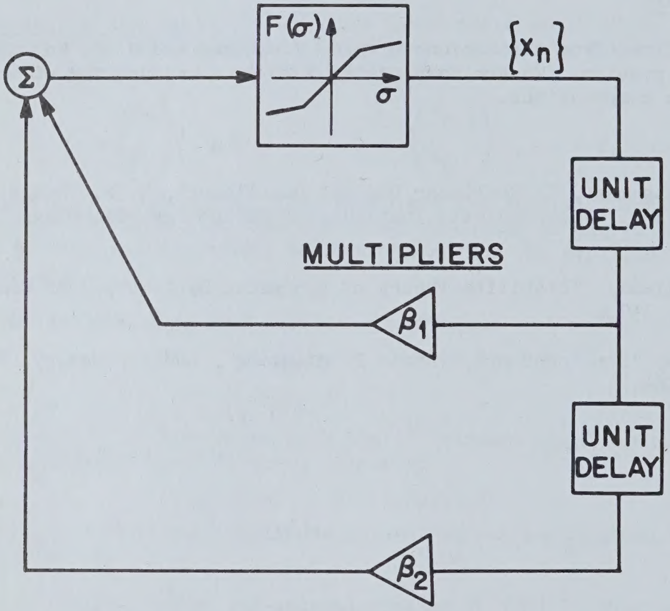

F(σ)
σ
{x_n}
Σ
UNIT DELAY
UNIT DELAY
MULTIPLIERS
β1
β2

FIGURE 1 THE DEVICE

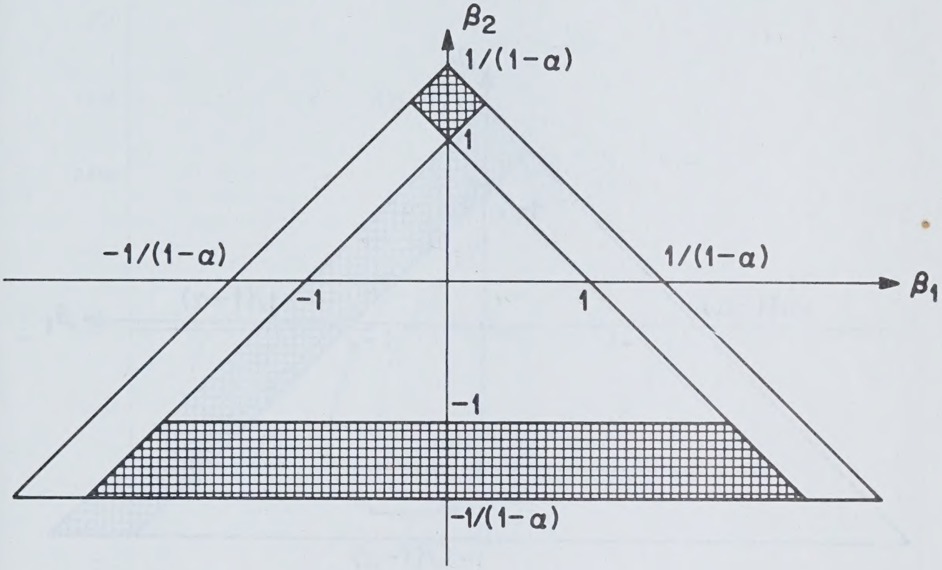

β_2
$1/(1-\alpha)$
1
$-1/(1-\alpha)$
$1/(1-\alpha)$
β_1
-1
1
-1
$-1/(1-\alpha)$

FIG 2 STABILITY REQIREMENTS ARE SATISFIED IN CROSS-HATCHED REGION

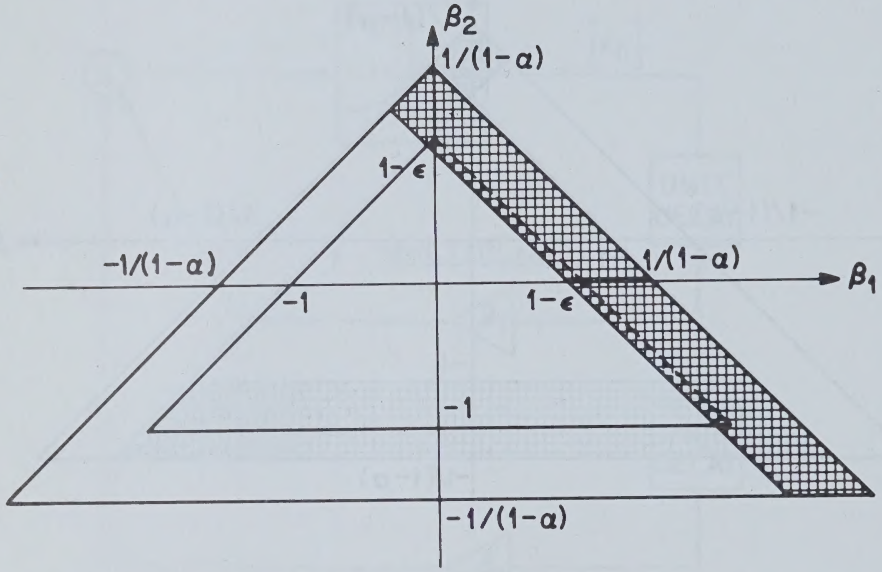
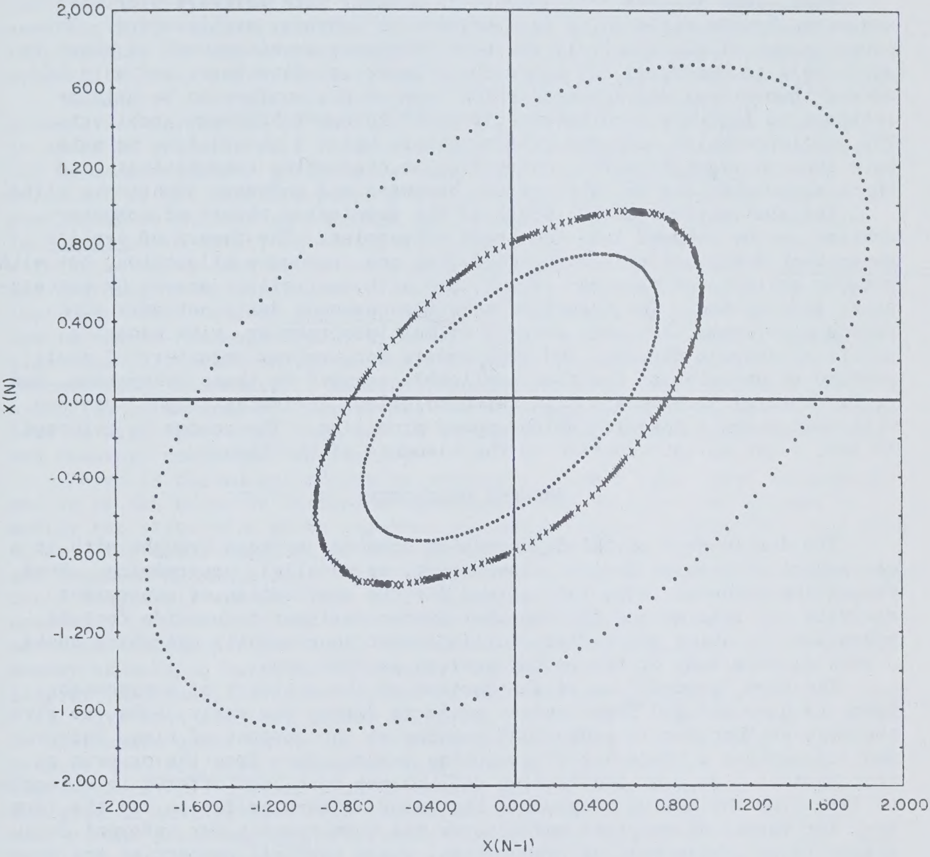

β_2
1/(1−α)
1−ε
−1/(1−α)
−1
1/(1−α)
β_1
1−ε
−1
−1/(1−α)

FIGURE 3 POSSIBLE STATIONARY POINTS IN CROSS-HATCHED REGION

FIGURE 4

OSCILLATION REGION AND LIMIT CYCLE



THE IMPACT OF THEORY ON SYSTEMS DESIGN

PETER J. DENNING
Department of Computer Sciences
Purdue University
W. Lafayette, Indiana 47907

INTRODUCTION

What, some readers will wonder, is a paper with software overtones suggested by its title doing in a session on computer architecture? This paper is out of place only if the term "computer architecture" is meant to apply only to hardware, but never to software, architecture; and only if recent theoretical developments which seem on the surface to be primary interest to software architects will never interest hardware architects. The developments in computer systems theory which I should like to note here concern algorithms for controlling or evaluating computations, and these algorithms are of interest to hardware and software architects alike.

The abstractions at the basis of the developing theory of computer systems can be grouped into two broad categories. The theory of process management deals not only with scheduling and resource allocation, but with process control and synchronization, and with controlling access by processes to information. The theory of memory management deals not only with paging algorithms, but with control of multiprogramming, with managing auxiliary storage devices, and with models for program begavior. I shall present an overview of the more applicable results in these categories, and review several problem areas in which solutions are being sought together with some current research which appear promising. The reader is referred to Ref. 1 for an introduction to the elements of the theory.

PROCESS MANAGEMENT

The development of third generation computer systems brought with it a new aspect of program design: concurrent, or parallel, programming. Even though the hardware technology allows for the realization of concurrent activity, it remains for the computer system designer to provide control operations by which programmers may implement concurrently executing tasks. A considerable body of theory has evolved in this area.

The term "process" as an abstraction of the activity of a processor seems to have emerged from several projects during the early 1960s, to give the concept "program in execution" meaning at any instant of time, independent of whether a processor is executing instructions from the program at that instant. So many conflicting definitions have been offered in attempts to formalize the notion of process that many writers prefer to use the term task for formal discussions and reserve the term process for informal ones. A task is an atomic unit of computation, whose terminal properties are specified but whose internal operation is of no interest in a discussion. Depending on the context, a task can be an instruction, a program, or even an entire system.

Process Coordination

Four important task control problems have come under considerable theoretical scrutiny. These problems can be defined formally in terms of a system of tasks interacting in specified ways [1].

First is the determinacy problem: the result of a computation performed by a system of tasks or common data is to be independent of the relative speeds of the tasks. Although many types of determinacy and determinate task systems have been studied, the primary result of practical significance is simply stated: A sufficient condition for the determinacy of a system of tasks is, for each pair of tasks which can be active simultaneously, the data which can be modified by the one must be disjoint from data which can be read or modified by the other. For many types of task systems of practical interest, this condition may also be necessary if the tasks are *uninterpreted* - i.e., only the control structure, but not the functions of the tasks themselves, is specified. The question of determinacy, or "speed-independence", has long been known to hardware designers, being studied in connection with asynchronous logic circuits in the late 1950s by Muller[2]; yet the applications of these results to programming were not fully appreciated for nearly a decade. Other important works on this problem are the asynchronous nets of Petri[3], the parallel program schemata of Karp and Miller[4], and the program graphs of Rodriguez[5].

Second is the *deadlock problem*. The following situation is to be avoided: A collection of parallel tasks competing for the same resources is stuck in circular wait, each waiting for at least one other to release its hold on resources while being unwilling to release any of its own. Solutions to the deadlock problem are of three types[6]: Preventing it by precluding in the system's design any or all of the conditions necessary for deadlock to exist; detecting it by texts on the system's resource-allocation state (and of course attempting recovery if deadlock exists); and avoiding it by continuously controlling the resource-allocation state so that only "safe" states are ever entered. The first solution to this problem in single-resource-type systems was given by Dijkstra as his "banker's algorithm"[7]; solutions in multiple-resource systems were studied by Habermann[8], Havender[9], Holt[10], and Murphy[11]. With the possible exception of Habermann's algorithm, the avoidance algorithms generate overhead so high that using them may do more harm than resorting to detection and recovery techniques instead[6].

Third is the *mutual exclusion problem*: Certain tasks must be programmed so as not to be in execution simultaneously, because they attempt to modify the state of a given resource and might otherwise interfere with each other. The first solution to this problem was offered by Dijkstra[7,12] and improved by Knuth[13]. These solutions were too complex to be practical; their complexity stemmed from their assumption that the only indivisible operation* provided by the host system was referencing memory, and the solution in effect transformed the mutual exclusion implemented by the memory addressing hardware into mutual exclusion at the software level. Simpler solutions, based on the use of indivisible *lock*/*unlock* instructions [14], have been studied; mutual exclusion of critical sections of code can be programmed quite easily if the hardware implements these instructions. Because the *lock*/*unlock* approach involves the "busy form of waiting", improvements were devised that allowed a processor to be preempted and assigned to another task when it encountered a locked critical section. Most of the solutions are related to the one proposed by Dijkstra[7], involving *semaphores* and the indivisible operations *wait* and *signal* (P and V, respectively) which would manage queues of tasks waiting for the semaphore value to become nonnegative. Habermann has developed a proof schema for establishing the correctness of instances of *wait* and *signal* in systems of tasks[15].

Fourth is the *synchronization problem*: There often is a requirement that a task be stopped at a given point in its execution unless and until a given event under the control of another task has occurred. Mutual exclusion is an example, but so are precedence constraints, ready / acknowledge signalling, message sending and receiving, and speed synchronization of cyclic tasks[1,15]. The wait and signal operations can be used to express the solution to all known synchronization problems, though not always in the most efficient possible way; for this reason other synchronization schemes have been proposed which are more efficient in given contexts (cf. The Multics Interprocess Communication Facility[16], or the RC4000 message-sending facility[17]). Many software architects would find their tasks greatly simplified if hardware architects were able to implement the requisite indivisible synchronization operations for them.

---

*A given set of operations is indivisible with respect to a given system if a) the action of any operation in the set appears as one step in the system, and b) if two processes attempt simultaneously to execute operations from the set, those executions will in fact occur in some order, one at a time.

Scheduling and Resource Allocation

Since algorithms for scheduling tasks on processors are often implemented in software, their interest has traditionally been restricted to software architects. However, two lines of recent research are producing results of potential interest to hardware architects.

One is deterministic multiprocessor scheduling, which seeks to compute minimal-length schedules for given partially-ordered sets of tasks with known execution times. The results about finding optimal schedules have been applied to exploiting parallelism among hardware function units, particularly in arithmetic expression evaluation[18]; other results of this type may ultimately be useful to hardware architects attempting to exploit parallelism and even to design buffers[19].

The second line of current research manifests itself as a rise in interest in queueing networks. A queueing network comprises: 1) a set of service centers, each implementing a specific queueing policy for the use of one or more units of some resource type and having a given distribution of service times, and 2) a transition matrix giving the probability that a customer moves from one service center to another upon completion of service. It has been discovered that, even if all the service centers of a given network are approximated with exponential distributions whose means are identical to those of the true distributions, the networks often can predict with remarkable accuracy the systems which they are intended to model[20-24]. This suggests that it is more important for the structure of the model (i.e., the job flows among service centers) to reflect those of the real servers. The analysis of these networks for exponential service distributions is straightforward[24], and of immediate interest to hardware architects interested in designing or evaluating networks of devices and function units.

Access Control

The problems of controlling access to information while conforming to stated security requirements are so important that a considerable amount of recent research has been devoted to abstracting the problems, formulating theories of protection and access control, and interpreting the theory in terms of systems design. It is being discovered that security measures must be included as part of the basic system design, and that considerable hardware support is required for efficient, correct operation. The theory is young and in need of improvement, but it is significant that independent research groups have arrived at surprisingly similar models for the problem (e.g.,[25] and [26]).

It has been proposed that the central abstraction of a protection system be a security, or access, matrix A. An entry A[D,X] gives the access privileges a task has toward object X when it is operating in protection domain D. Hardware support is essential to guarantee that the matrix A is used properly and efficiently[26]. It appears that many of the hardware techniques now used to render virtual memory implementations efficient can be extended to render access control mechanisms efficient, and that the extent of new hardware to support access control need not be great [26,27]. Hardware architects have a significant contribution to make in this area.

MEMORY MANAGEMENT

The memory management problem -- i.e., deciding how to distribute information among the levels of a memory and when to move it about -- has always been a problem of first importance. The objective of memory management policies usually is providing fastest possible execution speed by maintaining in main memory those parts of a program with the greatest likelihood of being referenced. The research in the theory of memory management is progressing along several lines, which are described in the sections following.

Most of the major results to date have been expressed in terms of paging, which admittedly sacrifices some generality in the analyses. Nevertheless, one could reformulate the analyses using variable block sizes together with a probability distribution over them, and would find that the results would not change significantly in nature. Moreover, the structural simplifications inherent in fixed-block-size allocations have been found so attractive that paging in some form or other can be found in the hardware of most systems today. The resounding success of the cache memory organization (e.g., IBM 360/85 or 360/195) further establishes that paging is not a barrier to highly useful systems, as some of its opponents might want us to believe.

Demand Paging

With respect to the structural properties of demand paging, the important paper by Mattson et al.[28] introduces the concept of a *stack algorithm*. Suppose $r(1)\ r(2)\ \ldots\ r(t)\ \ldots$ is a reference string over n pages, and is being managed in a memory space of m page frames, where $1 \leq m \leq n$. A demand paging algorithm is a stack algorithm if there exists a sequence $\underline{s}(0)\ \underline{s}(1)\ \ldots\ \underline{s}(t)\ \ldots$ of vectors (stacks), where $\underline{s}(t)=(x_1, \ldots, x_n)$ is a permutation of the page-names such that the memory contents at time t are $\{x_1, \ldots, x_m\}$ -- i.e., the top m entries in the stack. The *stack distance* $d(t)$ of reference $r(t)$ is its position in stack $\underline{s}(t-1)$; a page fault evidently occurs whenever $d(t)>m$. By maintaining n counters $c_1, \ldots, c_n$, corresponding to the various possible stack distances (such that c_j counts the number of occurrences of the event $d(t)=j$), it is possible to compute efficiently the entire curve of page faults $F(m)$ V. memory size m in a single pass of the reference string, since $F(m)=c_{m+1}+\ldots+c_n$. Mattson et al. also introduce the concept of *priority algorithm* (a model of certain demand paging algorithms whose replacement rule can be stated in terms of priorities on the pages) and show that every priority algorithm is a stack algorithm. The importance of this work is its demonstrating that a large number of paging algorithms of practical interest fall within one, well-balanced, highly-structured, easily measured and evaluated class.

Program Behavior

A phenomenon which has been observed in all empirical studies of program performance[29], and recently has become the subject of modeling, is *locality*. Locality, which is generally regarded as a fundamental characteristic of program behavior, is the property that a program tends to favor a subset of all its pages during any interval of execution, and membership in the favored subset tends to change slowly. *Locality models* for program behavior[30] are based on the idea of specifying probabilistic mechanisms for generating a sequence {L(i), T(i)} in which L(i) is a *locality* (a subset of pages) and T(i) a holding time in the locality. During the interval in which L(i) holds, only the pages from L(i) may be referenced. Under general assumptions about the mechanisms for generating L-T sequences, one can show that the rule *keep the current locality in main memory* is optimal; in fact all known optimal paging algorithms can be stated in terms of this rule framed in an appropriate locality model[30-32].

The most important locality model studied to date is the *LRU-stack model*[32,33], which derives from the stack algorithms. An LRU (least recently used) stack $\underline{s}(t)=(x_1, \ldots, x_n)$ is a vector listing the pages in order of decreasing recency of use; x_1 is in particular the most recently referenced page. (These stacks are so named because they are the priority lists used by the LRU paging algorithm.) A string of distances d(1) d(2) ... d(t) ... is generated according to some Markov chain, and is used to generate a sequence of references r(1) r(2) ... r(t) ... as follows. Let $\underline{s}(0)$ be an initial LRU stack, and suppose r(1) ... r(t) and $\underline{s}(1)$ ... $\underline{s}(t)$ have been generated. If d(t)=i, then the page at position i in $\underline{s}(t)$ is chosen to be r(t+1); then $\underline{s}(t+1)$ is generated by moving the page r(t+1) into position 1, and pushing the pages previously in positions 1,...,i-1 down one position. A locality L(t) at time t of any size k can be defined as the topmost k elements in the LRU stack $\underline{s}(t)$. The most appealing features of these models are: 1) locality properties can be emphasized to almost any degree desired by biasing the probability distribution of stack distances toward short distances, and 2) the LRU stacks observed while processing a reference string are precisely those internal to the model, and thus internal locality can be measured exactly by an external measurement, so that 3) the model can be used to predict an optimal set of pages to place in an available memory space. Stack algorithm techniques, therefore, hold out great promise of yielding useful models for program behavior.

Although much of the discussion of program models treats optimal memory management rules, the purpose is not to propose new memory management rules; it is instead to discover criteria for deciding which of a program's pages ought to reside in the space which has been allocated. By doing this, and by predicting locality for the immediate future, these models can be applied in dynamic control of multiprogramming. And by enabling efficient artificial generation of reference strings with given statistical properties, these models can be applied to systems evaluation.

Auxiliary Memory

Since most auxiliary memory device access times are significantly slower than main memory access times, these devices tend to become congestion points in a system. The policies of managing these devices, therefore, are concerned primarily with maximizing their throughput by eliminating needless idle time.

The paging drum, whose use in paging systems is widening, has been studied extensively. If requests to use the drum are sorted into queues, one for each drum sector, and if the drum services the head of a given queue as the corresponding sector rotates beneath the read-write heads, the drum will behave according to a shortest access time first (SATF) policy. Since an SATF policy minimizes total accumulated latency time, it is optimal with respect to maximizing throughput. The characteristics of paging drums are well understood[34,35].

Attempts to analyze the behavior of moving-arm disks have not been as successful. The problem is that an SATF policy -- which in practice reduces to a shortest seek time first policy -- tends to discriminate heavily against requests for the inner and outer cylinders. Although the throughput is maximum, system performance may suffer because certain requests must wait unduly long. A variety of disk arm-scheduling policies have been proposed and studied, the primary result being that no one policy is suitable over the entire range of load conditions[36]. Very little has been published about the effects of data organization on disk performance.

Multiprogramming

Results dealing with multiprogramming tend to be more sparse. There are no results about "optimal" multiprogramming, apparently because no one cost function seems to satisfy any or all of the requirements variously imposed on multiprogramming systems. The stack algorithm results do not extend to any useful characterization of multiprogramming management algorithms, principally because they deal with the behavior of paging algorithms for an arbitrary given reference string, whereas multiprogramming strategies must normally be designed to work only with probability assumptions about reference strings. In their application to program models, however, stack algorithm techniques may provide useful predictors for memory demand and so find application as components of control mechanisms.

Wilkes has provided a useful viewpoint about multiprogramming which shows the nature of the control problems inherent, the critical points at which control must be applied, and the range of applicable strategies and the feedbacks among them[37]. He imagines that each task is on one of three lists: the waiting list, where it has not been accepted into the system; the accepted list, where it is under consideration for receiving processor and memory resources; and the active list, where it is eligible to receive the services of a processor and to have pages in memory. Six policies must exist:

1. admission of tasks from waiting to accepted list;
2. admission of tasks from accepted to active list;
3. return of tasks from active to accepted list;
4. assignment of processor to tasks on active list; PROC. SCHEDULING
5. allocation of pages from the reserve to tasks on active list; PLACEMENT
6. deallocation of pages from tasks on active list back to reserve. REPLACEM RU

These policies are not independent and, in fact, inadequate control and coordination among them can lead to serious instabilities (e.g., thrashing[38]). The notion of a working set (an estimate of each program's current locality[39,40]) is important in implementing policies (5) and (6); it can be used in policy (2) as well, as a method of judging when memory is or is not fully committed. As our understanding of the dynamic control of multiprogramming increases the talents of hardware and software architects will be called into play to design the control procedures.

The working-set principles of memory management suggested above defines a variable partition strategy for multiprogramming, each active task being granted an allocation of space equal to its current working set size. In contrast, one can define a fixed partition strategy, according to which each task is granted an allocation invariant in size. Coffman and Ryan have shown that the variable-partition strategy is inherently superior to the fixed-partition strategy[41]. Implementing such strategies -- and so accruing their benefits -- will require more hardware support than many systems presently provide.

CONCLUSION

I have attempted to present here an overview of the development, origins, and applications of theory concerning computer system design and analysis. The results I have cited fall into the categories process management and memory management. A host of important systems problems have yet to yield to theoretical resolution, and there is at present very little consensus on what theoretical approaches are reasonable; these include the problems of performance measurement, reliability, systems design, data-base system design, semantics of information sharing, systems correctness, and so forth.

For the most part, the developing theory of computer systems deals with algorithms for doing things. It is ignorant of the separate existence of hardware and software architects. The solution procedures the theory gives us will very likely require a great deal more cooperation between hardware and software architects if they are to be implemented efficiently, or at all.

REFERENCES

1. Denning, P. J. "Third generation computer systems." Computing Surveys 3, 4 (Dec 1971), 175-216.

2. Muller, D. E. and W. S. Bartky. "A theory of asynchronous circuits." Proc. Int'l Symp. on Thy. Sw., Annals Comp. Lab. Harvard U. 29, Part I (1959), 204-243.

3. Petri, C. A. "Communication with automata." Suppl. 1 to RADC-TR-65-377, Vol. 1, Griffis AFB, Rome, NY (1966). [Original in German, U. Bonn (1962)].

4. Karp, R. M. and R. E. Miller. "Parallel program schemata." J. Comp. & Sys. Sci 3, 2 (May 1969), 147-195.

5. Rodriguez, J. "A graph model for parallel computation." MIT Project MAC report MAC-TR-64 (1969).

6. Coffman, E. G., M. J. Elphick, and A. Shoshani. "System Deadlocks." Computing Surveys 2, 3 (June 1971), 67-78.

7. Dijkstra, E. W. "Cooperating sequential processes." In Programming Languages (F. Genuys, Ed.), Academic Press (1968).

8. Habermann, N. "Prevention of system deadlock." Comm. ACM 12, 7 (July 1969), 373-377.

9. Havender, J. W. "Avoiding deadlock in multitasking systems." IBM Sys. J. 7, 1 (1969), 74-84.

10. Holt, R. C. "Deadlock in computer systems." Ph.D. Thesis, Cornell University, Department of Computer Science (1971): Report TR-71-91.

11. Murphy, J. "Resource allocation with interlock detection in a multi-task system." AFIPS Conf. Proc. 33 (1968 FJCC), 1169-1176.

12. Dijkstra, E. W. "Solution of a problem in concurrent program control." Comm. ACM 8, 9 (Sep 1965), 569.

13. Knuth, D. E. "Additional comments on a problem in concurrent programming". Comm ACM 9, 5 (May 1966), 321-323.

14. Dennis, J. B. and E. C. Van Horn. "Programming Semantics for multiprogrammed computations." Comm. ACM 9, 3 (Mar 1966), 143-155.

15. Habermann, N. "Synchronization of communicating processes." Comm. ACM 15, 3 (Mar 1972), 171-176.

16. Organick, E. I. The Multics System: An Examination of Its Structure M.I.T. Press (1972)

17. Brinck-Hansen, P. "The Nucleus of a multiprogramming system." Comm. ACM 13, 4 (Apr. 1970), 238-241.

18. Ramamoorthy, C. V. and M. J. Gonzales, Jr. "Subexpression ordering in the execution of arithmetic expressions." Comm. ACM 14, 7 (July 1971), 479-485.

19. Graham, R. L. "Bounds on multiprocessing anomalies and packing algorithms." Proc. AFIPS Conf. 40 (1972 SJCC), 205-218.

20. Baskett, F. "Mathematical models of multiprogrammed computer systems" TSN-17, Computation Center, Univ. Texas at Austin (Jan 1971).

21. Baskett, F. "The dependence of computer system queues upon processing time distribution and central processor scheduling." Proc. 3rd ACM Symp. on Oper. Sys. Princ. (Oct 1971).

22. Moore, C. G. III. "Network models for large scale time sharing systems." Univ. Michigan, Dept. Industrial Engrg, TR-71-1 (Apr 1971).

23. Buzen, J. "Analysis of system bottlenecks using a queueing network." Proc. ACM Symp. on Syst. Perform. Eval. (Apr 1971), 82-103.

24. Muntz, R. R. and F. Baskett. "Open, closed ,and mixed networks of queues with different classes of customers". UCLA, Dept. Computer Science report (Sept 1972).

25. Conway, R. W., W. L. Maxwell, and H. L. Morgan. "On the implementation of security measures in information systems." Comm. ACM 15, 4 (Apr 1972), 211-220.

26. Graham, G. S. and P. J. Denning. "Protection - Principles and practice." AFIPS Conf. Proc. 40 (1972 SJCC), 417-429.

27. Schroeder, M. and J. Saltzer. "A hardware architecture for implementing protection rings." Comm. ACM 15, 3 (March 1972), 157-170.

28. Mattson, R. L., J. Gecsei, D. R. Slutz, and I. L. Traiger. "Evaluation techniques for storage hierarchies." IBM Sys. J. 9, 2 (1970), 78-117.

29. Kuck, D. J. and D. H. Lawrie. "The use and performance of memory hierarchies - a survey." Software Engineering I (J. T. Tou, Ed.), Academic Press (1970), 45-78.

30. Denning, P. J. "On modeling program behavior." Proc. AFIPS Conf. 40 (1972 SJCC), 937-944.

31. Denning, P. J., J. E. Savage, and J. R. Spirn. "Models for locality in program behavior." Dept. Elec. Engrg., Princeton Univ., Computer Science report TR-107 (Apr 1972).

32. Spirn, J. R. and P. J. Denning "Experiments in program locality". Proc. AFIPS Conf. 41 (1972 FJCC).

33. Shedler, G. S. and C. Tung. "Locality in page reference strings." IBM Research report RJ-932 (Oct 1971).

34. Coffman, E. G. "An analysis of a drum input/output queue under scheduled operation in a paged computer system". J. ACM 16 .1 (Jan 1969), 73-90.

35. Denning, P. J. "A note on paging drum efficiency". Computing Surveys 4, 1 (March 1972), 1-4.

36. Teorey, T. and T. Pinkerton. "A comparative analysis of disk scheduling policies." Comm. ACM 15, 3 (March 1972), 177-184.

37. Wilkes, M. V. "The dynamics of paging" Computing Laboratory, University of Cambridge, Cambridge CB2 3QG, England.

38. Denning, P. J. "Virtual Memory." Computing Surveys 2, 3 (Sep 1970), 153-189.

39. Denning, P. J. "The working set model for program behavior." Comm. ACM 11, 5 (May 1968), 323-333.

40. Denning, P. J. and S. C. Schwartz. "Properties of the working set model." Comm. ACM 15, 3 (March 1972), 191-198.

41. Coffman and Ryan, 'A study of Storage partitioning using a mathematical model of Locality', CACM 15, 3(March 1972),

PERSPECTIVE AND STATUS OF LARGE SCALE DATA PROCESSING SYSTEMS

RICHARD E. MERWIN
SAFEGUARD System Office
Washington, D. C.

ABSTRACT

Following a brief introduction to possible approaches to the design of single and multiple data and instruction stream architectures, including pipelining, a description of the implementation of these architectural approaches in modern large scale computer systems is presented. A discussion of comparative operational performance of these systems concludes the paper.

I. INTRODUCTION

The demand for larger and more sophisticated data processing systems to solve larger and more complex mathematical problems continues in spite of the rapid progress that has occurred in this technology in the last decade. According to Roberts [1] the increase in computational capacity since 1959 has been at the rate S=1.85**(t-59) where S is the relative computing power expressed in megabits per second and t is the last 2 digits of the calendar year. This performance gain has largely resulted from introduction of semiconductor circuitry during this period. At the same time applications requiring ever increasing data processing performance are proposed [2] and the quest for greater computational capacity continues unabated. The present situation finds the "state of the art" of logic circuitry approaching practical and theoretical limits [3] in switching speeds and for further improvements in system performance, designers have had to look at system architectural approaches involving parallelism and pipelining of computational resources.

While parallelism and pipelining appear to offer the possibility of considerable increases in computational performance, this is not achieved without restrictions with respect to the formulation of the problem to be solved. The ingenuity shown in specifying a mathematical algorithm in a form suitable for computation within parallel or pipelined architectures by advocates of this approach has been very impressive. The future of this approach to expanding the capability of large scale data processing systems will hinge largely upon the success with which machines now being completed employing this internal architecture can be applied to the large computational problems of the next decade.

In order to survey this field and describe the status of current developments some preliminaries are required. A brief introduction to the terminology and techniques of large scale system architectures will be followed by an analysis of parallel and pipeline organizations, including sensitivity to problem formulation as a criteria for selection of an approach to achieve greater computational thruput. Next, the current status of large scale systems will be covered starting with the two machines considered "top of the line", i.e. the CDC 7600 and IBM 370 MOD195, followed by a review of recently developed parallel and pipeline machines.

II. ARCHITECTURAL CONSIDERATIONS

To form a background for ensuing discussions a brief overview of the subject of computer architecture is in order. Assume initially a machine as shown in Figure 1 with a storage unit containing instructions and problem data along with an arithmetic-logical unit (ALU) and a control unit which directs the operational sequence of the ensemble. Note that the input/output function is ignored. Two information streams, i.e. instruction and data, can be identified in the operation of this unit. These streams are drawn from the storage unit and the combination of an instruction data stream and program data stream constitutes a program. The computer can thus be visualized as an instruction stream and a data stream processor and this configuration will be designated as a single instruction-single data stream (SISD) system. The concept of stream can have both a software and hardware connotation. In a multiprogramming environment several sets of instruction and data representing different programs can be resident in storage only one of which is being operated on by the ALU.

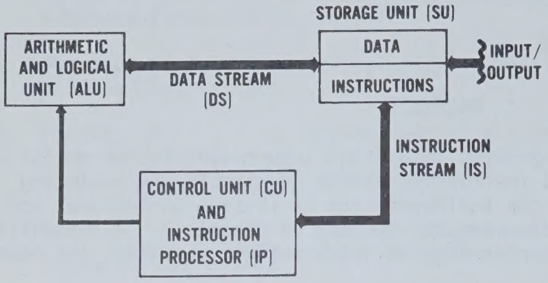

STORAGE UNIT (SU)
ARITHMETIC AND LOGICAL UNIT (ALU)
DATA
INSTRUCTIONS
INPUT/OUTPUT
DATA STREAM (DS)
INSTRUCTION STREAM (IS)
CONTROL UNIT (CU) AND INSTRUCTION PROCESSOR (IP)

FIGURE 1 - BASIC MACHINE ORGANIZATION

The definition of an instruction and data stream leads naturally to the possibility of their replication, which suggests the three additional architectural classifications shown in Figure 2.

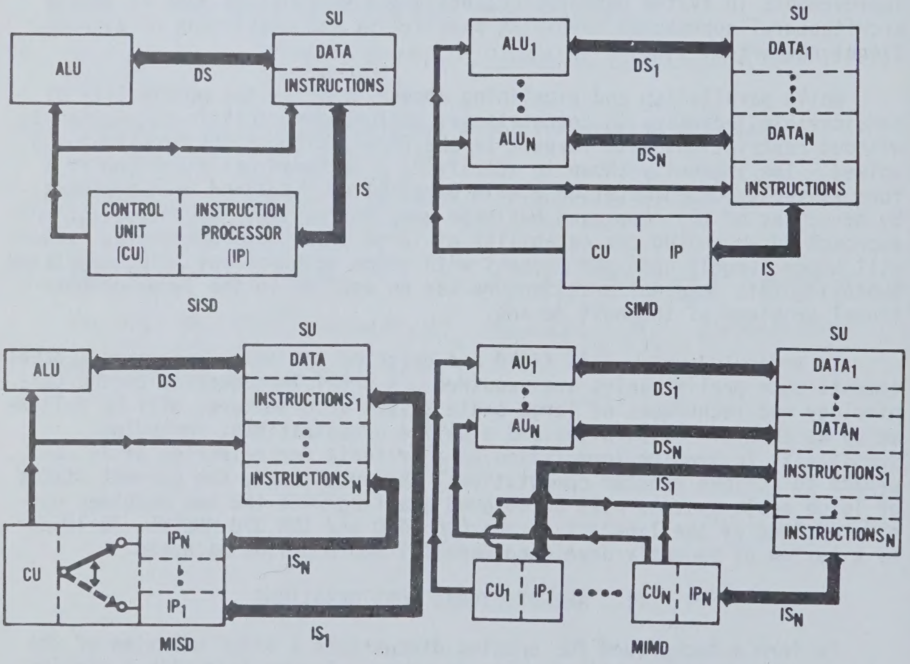

SU
ALU
DATA
INSTRUCTIONS
DS
IS
CONTROL UNIT (CU)
INSTRUCTION PROCESSOR (IP)
SISD
SU
ALU_1
ALU_N
DS_1
DS_N
$DATA_1$
$DATA_N$
INSTRUCTIONS
CU
IP
IS
SIMD
SU
ALU
DS
DATA
$INSTRUCTIONS_1$
$INSTRUCTIONS_N$
CU
IP_N
IP_1
IS_N
IS_1
MISD
SU
AU_1
AU_N
DS_1
DS_N
IS_1
$DATA_1$
$DATA_N$
$INSTRUCTIONS_1$
$INSTRUCTIONS_N$
CU_1
IP_1
CU_N
IP_N
IS_N
MIMD

FIGURE 2 - FOUR BASIC ARCHITECTURAL CONFIGURATIONS

These configurations are designated single instruction-multiple data stream (SIMD), multiple instruction-single data stream (MISD), and multiple instruction-multiple data stream (MIMD). The four computer architecture categories so defined are basic but not represented in a pure form in any computer in existence today. However, most existing computers can be assigned, from an organizational viewpoint, to one of these categories with varying degrees of fidelity. A detailed discussion of these architectural classifications is presented by Flynn [4].

Two techniques which are commonly employed to augment computer performance are overlap of instruction processing with ALU processing and overlap of instruction and arithmetic operand storage fetch times with instruction and arithmetic processing cycles. For many instructions, e. g. branch or load register, no arithmetic operation is required so these functions can be carried out concurrently with the ALU operation which maximizes the utilization of the arithmetic hardware. Instructions and operands can be prefetched and stored in buffers in the ALU which eliminates the wait time for storage access cycles. Both these techniques are widely used in large scale data processors along with a high speed local storage unit, often called "cache", which provides storage access times approximately equal to the ALU basic cycle times. All of these techniques along with the current semiconductor circuit switching speeds of several nano seconds (ns.) leads to instruction processing times of from 25 to 75 ns. which implies instruction processing rates of from 10 to 30 million instructions per second (MIPS). Instruction processing rate is only meaningful for SISD architectures and where multiple data streams are being processed a more meaningful term i.e. megabits per second (MBS) introduced above, is used to represent the input rate of information being fed by all the data streams into the processing units. Another advantage of the MBS rating is that it includes the data word length as a parameter and hence is more indicative of the total processing capability.

The above discussion ignores many considerations including contention among the instruction and arithmetic processors trying to access the same storage unit along with data and instruction precedence requirements which can negate overlap procedures. These factors are called turbulence and an analysis of their impact on system performance is given by Flynn (4). In order to achieve the processor speeds noted above the arithmetic units are internally pipelined as described by Hallin et al (5). This technique in conjunction with the buffer registers for arithmetic operands implies that the data for several instructions is being simultaneously operated upon by the ALU and makes these units particularly susceptible to interactions between the instruction processor and data stream, and precedence relations between the operands.

Starting with the IBM 7030 and with the CDC 6600 and 7600 it has become common practice to include peripheral processors to off-load such functions as I/O control, operating systems, and teleprocessing functions from the ALU. To simplify the following discussion, these devices will not be considered but it should be recognized that they play a significant role in system performance.

III. PERFORMANCE MEASUREMENT AND ANALYSIS

Performance measurement and analysis of large scale data processing systems still lacks the precision to be called a science. The approaches to this problem vary from comparing instruction and bit processing rates, execution of benchmark tests combined with simulated system components as drivers, and use of sophisticated mathematical techniques to analyze system behavior. The accuracy of evaluation of a machine's performance by these techniques can be precise but the relation of the projected performance to actual application experience is often nebulous and leads to poor estimates as to the suitability of a candidate computer for a particular application.

Computer performance measures can be broadly categorized in two ways, one associated solely with hardware performance and the other involving the whole system environment including software. Hardware measures include such parameters as basic ALU cycle time (in ns.) and word length (in bits) storage capacity (in words) and input/output capacity (in MBS). The reciprocal of the basic ALU cycle (assumed to be a basic instruction cycle for add) gives the upper bound on instruction processing rate (MIPS). Typically in a SISD machine the useful MIPS rating lies in the range of 1/4 to 1/3 the maximum value and this figure times the word length in bits gives the MBS rating. Another popular method for estimating the useful hardware MIP rating based upon an instruction mix associated with the application is the "Gibson Mix" which is a weighted average of instruction processing rates whose reciprocal is the expected ALU performance in MIPS on a particular job. The main virtue of these performance ratings is that they are easily understood and meaningful to management personnel who are involved in selection of computer equipment.

When software is to be included in the performance measurement procedure a benchmark test based upon the application for which a computer is being selected is the most frequently used technique. Another approach to this problem includes hardware and system simulators and combinations of these techniques represents the most accurate means for estimating hardware and software performance in a particular environment. A problem with this latter approach is the expense of running the simulations and benchmarks and it is a time consuming procedure . Even with extensive simulation and benchmark tests,there is often a serious lack of correlation between benchmark measurements and actual operation experience. This problem stems primarily from software effects which can be attributed to the use of programming languages for software production. It has been shown by Henriksen et al [6] that as much as a two to one reduction in system performance can be expected in software coded in FORTRAN as compared to the same programs coded in assembly language. When a benchmark test is included in a computer selection procedures, the manufacturers tend to put their best programmers to work coding the benchmark and achieve a level of performance considerably higher than that experienced by the buyer with conventionally coded applications. An excellent summary of available performance analysis methods is presented by Miller [7] along with annotated bibliography of the recent literature on this subject.

Analytic approaches to predicting system performance have been mainly concerned with attempts at analyzing specific operational environments such as multiprogramming [8,9] multiprocessing [10] and parallel processors. Of particular interest are the space time studies of parallel and pipelined (SIMD) architectures by Chen [11] where a simple geometric argument is used to show the basic equivalence of these two approaches to increasing systems thruput. A brief description of the essence of this analysis is of interest. Job space is shown in Figure 3 with the dimension of work and time. Job width W is normalized to the computational output available from a SISD machine assumed to correspond to one of M parallel processors and ignores all considerations associated with input-output, software effects and turbulence. The parameters ρ and η corresponding to the density of job space showing parallelism to total job space,and the efficiency of hardware

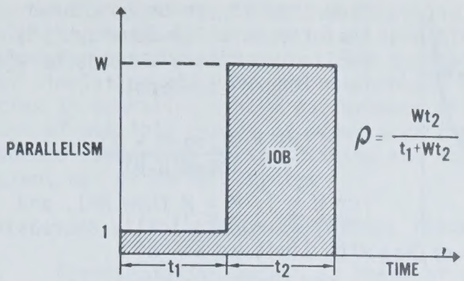


FIGURE 3 - SPACE-TIME DIAGRAM OF A STANDARD JOB (FROM CHEN [11])

utilization in execution of a job, respectively, are defined in terms of Figure 3* as:

$$\rho = \frac{Wt_2}{t_1+Wt_2}$$

$$\eta = \frac{Wt_2}{M(t1+Nt_2)}$$

Where N is the integer part of the expression (W+M-1)/M which corresponds to the number of sweeps that would be required for M processors to execute a job of width W. A plot of ρ versus η as shown in Figure 4* for an SIMD system with 32 processors clearly shows the price paid in efficiency of hardware utilization for systems with many parallel processors when the parallelism density drops below unity. By an ingenious argument Chen goes on to show that the analysis for job handling efficiency for a parallel architecture applies equally to pipelined systems. This leads to the conclusion that efficient application of the SIMD organization, especially from the hardware viewpoint, requires a density of parallelism in the job space of better than 0.9.

Users with an application where speed is of utmost importance would claim that parallel or pipeline architectures offer a speed advantage and with large scale integration (LSI) technology promising cheap fast small circuitry, efficiency of hardware utilization becomes of secondary importance. To examine this premise, Chen's results are extended slightly to consider the speed of a parallel processor as a function of job parallelism density ρ. The speed ratio π of a parallel to a sequential processor is, from Figure 3:

$$\pi = \frac{\text{sequential processor execution time}}{\text{parallel processor execution time}} = \frac{t_1+Wt_2}{t_1+Nt_2}$$

and letting $\beta = t_2/t_1$, π becomes

$$= \frac{1+W\beta}{1+N\beta} \quad \text{where W and N are as defined above.}$$

The parallelism density ρ in terms of β becomes

$$\rho = \frac{W\beta}{1+W\beta}$$

and solving for ρ and substituting in π gives

$$\pi = \frac{1}{1-\rho(1-\frac{N}{W})} = \eta M$$

Plotting π as a function of $\rho(W)$ for several values of W(N assumed equal to 1) as shown in Figure 5 again shows the precipitous drop in the ratio of speed of a parallel processor to a sequential processor as ρ falls below unity. It can be shown from the expression for π that

$$\rho = \frac{W(\pi-1)}{\pi(W-N)} \qquad (\Pi \leq W)$$

---

*Figures 3 and 4 are essentially reproductions of Figures 1 and 3 of reference 11.

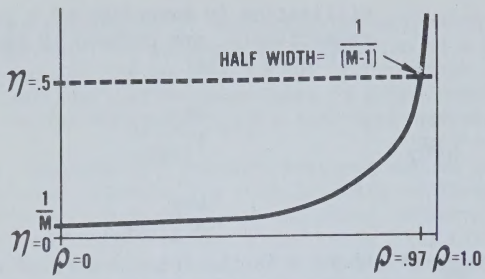


FIGURE 4 - EFFICIENCY vs. PARALLELISM DENSITY FOR M=W=32 (FROM CHEN [11])

From this it can be concluded that the parallel density (ρ^*) implied by Minsky's conjecture[13] ($\pi = \log_2 M$) becomes:

$$\rho^*(W) = \frac{W(\log_2 M - 1)}{\log_2 M(W-N)}$$

For $W \geq 1$, $M \geq W$ then N=1, and $\rho^*(W)$ is monotonically decreasing with limit as $W \to \infty$ of

$$\lim_{W\to\infty} \rho^*(W) = \frac{\log_2 M - 1}{\log_2 M}$$

which is a lower bound on $\rho^*(W)$, $1 \geq N \geq W$.

A plot of this relation in Figure 6 shows that a high density of parallelism is implied by Minsky's conjecture. It is expected that many of the above results relative to the speed improvement possible with a parallel processor would carry over to the pipeline processor case, however, pipeline startup procedures and obtaining expressions for equivalent sequential operation times greatly complicates the mathematical discussion.

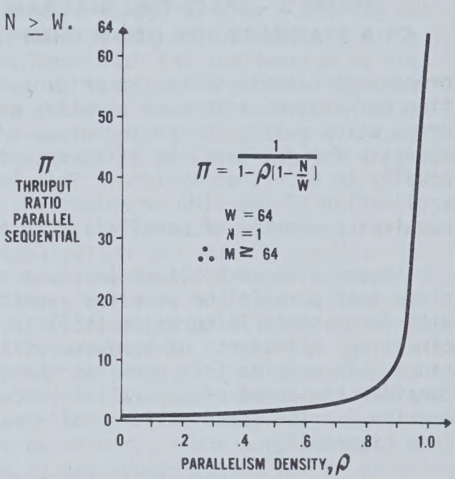


FIGURE 5 THRUPUT (π) RATIO VS. PARALLELISM

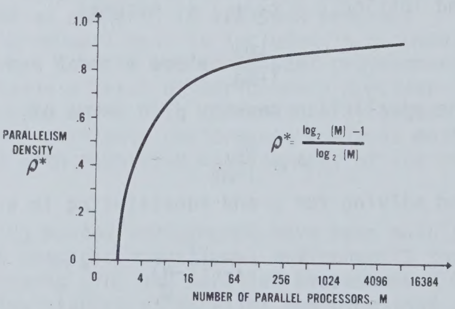


FIGURE 6 PARALLELISM DENSITY IMPLIED BY MINSKY'S CONJECTURE (LOWER BOUND)

Performance analysis of MIMD architectures consists of simulation studies and theoretical treatments of scheduling algorithms which are basically software controlled. An excellent overview of task scheduling studies is given by Manacher [12]. This includes a discussion of Graham's bound which states that the ratio of the run times ω and ω' for MIMD systems with N and N′ processors will be limited to:

$$\frac{\omega'}{\omega} \leq 1 + \frac{N-1}{N'}$$

due to changes in the order of execution of tasks, removal of some precedence relations between tasks, changing N, and shortening the run-time of some tasks. If N = N′ then the ratio of run-times reduces to

$$\frac{\omega'}{\omega} \leq 2 - \frac{1}{N}$$

While of academic interest, other factors not considered in the above analysis play an important role in MIMD performance evaluation including contention between processors accessing the same storage unit, and design and computational support of an operating system that achieves an efficient use of processor resources. The approach to solving these problems is use of simulation and hardware performance monitors that point out inefficiencies in operating system performance and measure storage contention. Storage of multiple copies of often used routines, use of reentrant coding and careful attention to task timing and scheduling are required to make efficient use of an MIMD system.

IV. OPERATIONAL AND PERFORMANCE STATUS

Representative models of three of the four possible architectural approaches to large scale data processors have been constructed. Operational characteristics of a group of these machines are shown in Table I.

MACHINE	DATA INSTALLED	ARCHITECTURAL TYPE	BASIC WORD LENGTH BITS	PARAMETERS ALU CYCLE TIME NS	PERFORMANCE MIPS	MBS	COMMENTS
IBM 7090	1960	SISD	36	2(us)	0.23	8.2	Fairly pure example of SISD architecture
IBM 7030 (STRETCH)	1961	SISD	32	1(us)	0.5	32	Included I/O Processor and interlaced storage
IBM 360 MOD195	1969	SISD	32	54	10.0	640	Pipelined ALU & I/O Processor and cache storage (content addressable)
CDC 6600	1964	SISD	60	100	2.4	145	Includes peripheral
7600	1969	SISD	60	27.5	12.0	720	Processors & pipelined ALU, 2 levels of addressable core storage
CDC STAR	1973	SIMD	64	40	25.0	3200	Vector system
TI ASC	1972	SIMD	32	50	25.0	1600	" "
ILLIAC IV(64)	1973	SIMD	32	62.5	128.0	8200	64 Processor array
BTL CLC	1970	MIMD	32	200	15.0	960	Pure Multiprocessor

TABLE I Operational Characteristics of Past and Present Large Scale Data Processors

The CDC 7600 and IBM 370 MOD 195, both SISD systems, have been available since 1970 and have been competitively performance measured many times. Processing capability of both systems is extremely sensitive to the quality of the software with performance ranges of as high as five to one being attributable to this factor. Extensive benchmark tests in which software factors have been minimized to the maximum extent possible give the CDC 7600 abouta 20%edge in performance over the IBM 370 MOD 195 which would be expected in view of the almost two to one ratio between basic ALU cycles (27.5ns. versus 54ns. respectively). The importance of the software on the performance of these systems is illustrated by the experience in a comparative test involving application oriented benchmarks with simulated drivers carried out in 1971. Load handling capability of both systems was almost doubled (i.e. one additional 7600 and MOD 195) by fine tuning software and optimization of what had been rigidly specified software algorithms.

Aside from an IBM 370 compatable Central Processing Unit (CPU) being developed using large scale integration (LSI) components by Amdahl Corporation, there appears to be no attempt to further increase the processing thruput of the currently available SISD systems. CDC in order to meet requests for greater thruput have offered the CDC 7700 which is an MIMD system. There has been some speculation about systems with basic ALU cycle times in the range of 10 to 20 ns. but such systems are probably geared to the development of LSI technology to support such performance levels. Further the impact of other architectural approaches, particularly SIMD, and the optimum utilization of LSI must be evaluated and tested before the final trend in future developments of SISD machines can be determined.

For applications which require more processing capability than available in SISD machines, consideration is being given to ordering SIMD systems with the most interest currently in the vector machines. Two such systems, the CDC STAR and Texas Instrument Corp (TI) ASC, have been ordered by the AEC, Weather Bureau, and U.S. Army for large scale applications. Both of these systems were to have been operational in 1971 but due to technical difficulties with both components and logic design these systems are just now becoming fully operational. Some design specifications haven't been fully met and no significant benchmarks have been carried out. Both of these systems were to have been included in the large benchmark tested noted above and preliminary software was developed but neither system was operational at the time and only estimates could be made based upon data available from partial tests of the software made on these systems. For the TI ASC these results looked encouraging and seem to indicated that this system will offer a significant increase in thruput over available SISD systems for the benchmark problems studied.

The parallel processor implementation of the SIMD architecture is in a much earlier stage of development and only estimates are available for operational performance. The two main efforts in this area are the now famous ILLIAC IV designed by the University of Illinois for ARPA which is being installed at the NASA Ames Laboratory and the Parallel Element Processing Ensemble (PEPE) designed by BTL for the Army Ballistic Missile Defense Agency. A prototype has been constructed of this latter machine, which has now been taken over by the System Development Corporation (SDC), and was successfully tested at a 16 processor element level in the aforementioned benchmark tests. The ILLIAC IV is still not fully checked out and no operational experience is available although extensive studies carried out at the University of Illinois suggest a number of applications which are well suited to take advantage of this machine's capability. The degradation in hardware utilization predicted in the previous section on performance evaluation for SIMD architecture hasn't been adequately investigated upon this type of hardware as yet. As results from competitive benchmarks between SIMD and SISD systems become available an assessment of the performance of these parallel and pipeline systems will become known.

Commercial versions of the MIMD architecture have been available since 1966 and are mostly two or three processor systems (CDC 6500 and IBM 360 MOD65 MPL, UNIVAC 1108). Processing thruput ratios to the equivalent SISD system are generally quoted to be in the 1.5 to 1.8 range for two processor systems. When larger numbers of processors have been integrated in this configuration, progressive degradation in performance has been observed. This stems from two factors noted above i.e. the difficulty in developing an operating system which keeps a number of processors continuously busy and the inherent problem of storage contention between processors accessing the same storage unit. The most ambitious of the MIMD designs is the multiprocessor designed by BTL for a special real-time application where up to 10 processors can be simultaneously in operation under the control of a real-time operating system which schedules jobs in a predetermined time sequence to maximize processor utilization. A progressive degradation factor is assumed for storage contention based upon the number of processors in simultaneous use. Again as in the case of SIMD systems the full utilization of the MIMD architectural approach is just beginning with considerable interest being shown in the performance of early development models.

BIBLIOGRAPHY

1. Roberts, L.G. "Data Processing Technology Forecast" Private Communication.

2. Ware, W. H. "The Ultimate Computer" IEEE Spectrum March 1972.

3. Keyes, R. W. "Physical Problems and Limits in Computer Logic" IEEE Spectrum May 1969.

4. Flynn, M. J. "Some Computer Organizations and Their Effectness" IEEE Transactions on Computers Vol C 21, No. 9, September 1972.

5. Hallin, T. G., Flynn, M. J., "Pipelining of Arithmetic Functions" IEEE Transactions on Computers Vol C 21, No. 8, August 1972.

6. Henricksen, J. H., Merwin, R. E., "Programming Language Efficiency in Real-Time Software Systems" AFIPS Conference Proceedings (SJCC) Vol 40, Montvale, N. J. 1972.

7. Miller, E. F., "Bibliography on Techniques of Computer Performance Analysis" Computer Vol, No 5, September 1972.

8. Ramamoorthy, C. V., Chandy, K. M., "Optimization of Memory Hierarchies in Multiprogrammed Systems" JACM, Vol 17, No 3, July 1970.

9. Coffman, E. G., Muntz, R. R., Trotter, H. "Waiting Time Distributions for Processor-Sharing Systems", JACM Vol 17, No 1, January 1970.

10. Muntz, R. R., Coffman, E. G. "Preemptive Scheduling of Real-Time Tasks on Multiprocessor Systems" JACM, Vol 17, No 2, April 1970.

11. Chen, T. C., "Parallelism, Pipelining, and Computer Efficiency" Computer Design, June 1971.

12. Manacher, G. K., "Production and Stabilization of Real-Time Task Schedules" JACM, Vol 14, No. 3, July 1967.

13. M. Minsky and S. Papert, "On some associative, parallel, and analogue computations" in Associative Information Techniques, E. J. Jacks, Ed. N Y Elsevier, 1971.

MINICOMPUTER SYSTEMS DESIGN FOR REAL-TIME PATIENT MONITORING*

S. S. YAU, G. F. DEPALMA, AND G. W. DIETRICH
Departments of Computer Sciences and Electrical Engineering,
and Biomedical Engineering Center
Northwestern University
Evanston, Illinois 60201

ABSTRACT

The requirements of computerized patient monitoring systems are discussed, and the design of minicomputer systems for real-time patient monitoring is considered in general. The minicomputerized patient monitoring system currently being developed at the Biomedical Engineering Center of Northwestern University is presented as an example.

INTRODUCTION

One of the most sophisticated computer applications in patient care is in the area of real-time patient monitoring [1-6]. Because of the recent rapid development, many powerful commercial minicomputer systems are available, and these minicomputers have the advantages of low cost, ease in installation, operation and maintenance, and flexibility in tailoring them to various specific needs. In this paper we shall discuss the design of minicomputer systems for real-time patient monitoring, and then present the minicomputerized patient monitoring system currently being developed at the Biomedical Engineering Center of Northwestern University as an example. Before we proceed, let us first analyze the requirements imposed on computerized patient monitoring systems.

REQUIREMENTS ON COMPUTERIZED PATIENT MONITORING SYSTEMS

When we design a practical computerized intensive care patient monitorins system, the following factors should be considered: 1) capability of the system, 2) simplicity in its use, 3) its cost effectiveness, 4) installation requirements, 5) reliability, and 6) safety requirement. We shall discuss each of these factors.

1) Capability

The system is operating in real-time and must be capable of continuously and simultaneously monitoring all the patients' vital functions and identifying signals of medical significance, such as ECG arrhythmias, signs of shock from blood pressure, irregularities and trends in temperature, elevated CO_2 concentration, irregular respiration rates, etc. When the monitored signals of any patient in the intensive care unit deviate from the patient's expected normal values beyond certain threshold specified by the physician, the system must be able to issue a warning signal, such as an alarm or flash signal, to draw the attention of the medical personnel, and indicate the cause of this warning signal. The information gathered from the patients must be analyzed and the results, current and past, must be organized in a clear and consise form which can be displayed at the patients' bedside consoles upon request. It should be easy to enter information to the system by nurses and produce hardcopy as the patients' permanent file. The information entered includes the patients' medical history, medical personnel's comments, drug administration, medical treatment, as well as the results of off-line analysis of blood, gas, urine, etc. In order to be economic, the system must be able to monitor multiple patients, and the entering or releasing a patient from the intensive care unit must not interfere in the monitoring of other patients. The minimal capability of such a system is expected to perform the routine data gathering and recording functions, which otherwise constitute the most time-consuming portion of

---

*The work reported here was supported by Public Health Service Grant 5 P01 GM 15418.

nurses' work load in an intensive care unit. It is anticipated that any computerized patient monitoring system should perform at least such routine functions much more reliably and efficiently than nurses. A sophisticated computerized patient monitoring system may be able to provide medical diagnosis and prediction of patients' conditions, which can be used as valuable references for physicians.

2) Simplicity in its use

In order to have a computerized patient monitoring system easily acceptable to medical personnel, the system should be easy to use and provide efficient man-machine interaction. The instructions for users should be as simple as possible and no special training should be imposed on the medical personnel for using the system. For instance, all the input/output requests and entries may be made through a 10-digit keyboard at each bedside console with the requests coded, and pressing a special button will display all the request command codes at the bedside console. Thus, the user is not bothered with typing long requests according to a specific format in order to communicate with the system. Entry of patients' information into the system may be done at the central monitoring station. The system should be immune to all illegal or erroneous requests, such as erroneous codes. This implies that no user's requests may destroy a patient's file or harm any patient.

3) Cost effectiveness

To consider the cost effectiveness of a computerized patient monitoring system, we may make a simple comparison of the cost between this system and a regular nursing staff. For a regular intensive care unit, every pair of beds needs one nurse with three shifts a day (i.e., three nurses for a pair of beds). Of course, in such an intensive care unit, equipment for monitoring patients, such as transducers and ECG analyzers, etc., are still required. For a computerized patient monitoring system, the number of nurses can be largely reduced. Normally, only two nurses with three shifts (i.e. a total of six nurses) should be able to manage all the beds in a computerized patient monitoring system. The cost of such a system includes the transducers, a multi-station interactive display system, a computer system and interface cards and its operating and maintenance cost. With the low cost and increasing capabilities of current minicomputers, a computerized patient monitoring system using a minicomputer system can easily monitor eight or more patients simultaneously. Thus, it is not hard to conclude that such a system is more economic than a regular patient monitoring system with a large nursing staff, even if the advantages and great capability of the computerized system are not taken into account.

4) Installation requirements

Because of hospital regulation and the requirement on fast response time for an interactive system, the computer system of a computerized patient monitoring system should be separated from , but adjacent to, the intensive care ward. Since the space close to the intensive care ward in a hospital is always very valuable, the computer system must have low installation requirements, i.e., small physical size, low demand for electric power and cooling capacity. With many minicomputers now implemented using integrated circuits, it is easy to find satisfactory minicomputer systems having low installation requirements.

5) Reliability

One of the most severe requirements of a computerized patient monitoring system is its reliability. The system should be able to perform its minimal functions, such as data acquisition, certain analysis, and display, even if the computer system has faults. In the design of such a system, we should consider real-time fault detection for the computer system, and the system should have alternative paths for vital data so that it can degrade its operations gracefully in case of faults. Furthermore, spare parts for fast repair should be available in order to bring the system back to full operation level as quickly as possible. For this purpose, the number of spare parts of the computer system necessary for common types of repairs should be kept minimal. Because the system should operate continuously (24 hours every day), in the selection of the minicomputer system consideration should be given to the availability of fast services from the computer manufacturer if the hospital does not have the necessary personnel and facilities to maintain the computer system.

6) Safety requirement

Because the system is connected to critically ill patients, strictest safety requirements must be satisfied. The computer system and the computer room must satisfy all the electrical and fire codes. Since many current minicomputer systems have very low installation requirements, there should not be any difficulty in selecting a minicomputer system satisfying the safety requirements. However, great attention should be given to the safety features of transducers, preamplifiers and other equipment installed at the bedside consoles. This includes electrical leakage between the system and the patients, common ground, etc.

HARDWARE/SOFTWARE MINICOMPUTER SYSTEM DESIGN CONSIDERATION

In the design of computer systems for most special applications, it normally involves two parts: One is the selection of commercially available hardware and software which are best suitable to the needs, and the other is the addition of special purpose hardware and design of new software or modification of available software which are required to make the system operate efficiently. With the requirements for computerized intensive care patient monitoring systems described in the last section, we shall now discuss the design of minicomputer systems for such an application. In this section, our discussion will be focused on the consideration of selecting commercially available hardware and software, addition of special purpose hardware, and improvement of software.

1) Selecting commercially available hardware and software

Before considering the selection of the computer system, we should carefully analyze all the requirements imposed on the patient monitoring system. Among the requirements stated in the last section, the one having the largest influence on the selection of the computer system is obviously the expected capability of the patient monitoring system. If we expect that the system will only perform the normal routine functions of nurses in the intensive care unit, such as taking readings of patients' monitored parameters, then we only need a very simple computer. On the other hand, if we expect that the system has the capability of not only monitoring patients' parameters, but also providing certain sophisticated analysis and flexibility in man-machine intereaction, the computer system should be a more powerful one. Normally, the minicomputers considered for this type of application may range from 12-bit word to 18-bit word with 4K to 32K core memory and some kind of mass storage, such as disk or magnetic tape. While the selection of the computer system is affected by many factors, we can discuss some general guidelines in the selection of computers and their important options. Since the options may constitute a major portion of the computer cost, only the options which are essential to the operation of the system or which will increase the capability of the system should be ordered.

In selecting the minicomputer system for patient monitoring, we first have to decide whether we need a 12-bit or 16-bit (or 18-bit) machine. The differences between these two classes of minicomputers are obviously the cost and capability. If the capability of the patient monitoring system is expected to be extensive, we probably have to use a 16-bit (or 18-bit) machine; otherwise, a 12-bit machine may be sufficient.

After deciding which class of minicomputers should be used, we should evaluate the minicomputers in the same class according to the following areas:

- Cost
  - Equipment cost
  - Installation requirements: size, cooling, power
  - Maintenance cost & service availability
  - Operating cost
- Instruction Repertoire
- Memory Cycle Time
- Instruction Execution Time (in # memory cycles)
  - Memory reference instructions
  - Register reference instructions
  - I/O instructions
- Available Hardware Options
  - Hardware multiply/divide
  - Floating-point arithmetic
  - Power-fail-restart
  - Memory protect
- I/O Structure
  - DMA channels
  - Interrupt system
  - Ease of interface of I/O devices
- Available Peripherals
  - Real-time clock
  - Paper tape
  - Teletype
  - Disk
  - Magnetic tape
  - A to D converter
- Available Software
  - Disk operating system
  - Real-time executive
  - Assembler
  - Fortran

It is noted that some of the commercially available minicomputers may be eliminated from consideration immediately because of certain special restrictions, such as installation requirements, maintenance service availability, speed, cost range, etc. The final cost of the selected minicomputer system largely depends upon what options are included in the system. Except those options which are required for the operation of the system, such as the real-time clock, interrupt system, which are required for any real-time system, it is often not obvious to the designer whether many other important options should be included. In the following, we shall discuss whether some of these options are needed.

Among the available hardware options, power-fail-restart and memory protect should normally be included in the minicomputer system for patient monitoring. In a hospital where the voltage of the power supply may vary due to the use of other equipment which draws a large amount of power, the *power-fail-restart* is essential. With this option, the patient monitoring system may operate 24 hours a day without loss of programs and possibly with only minor loss of data entering the system during the faulty period. It also provides "turn-key" operation for the system which is not operating continuously. *Memory protect* provides fences against intruders into data or program areas of the main memory. During the testing stages of a newly developed computerized patient monitoring system with man-machine interaction, these fences can prevent a lot of grief as well as aid in debugging. It is noted that in some minicomputer systems these two options become standard items, while most minicomputers can have them as optional items with a nominal cost.

If integer multiply and divide operations are used frequently in the programs, it may be justifiable to have the option of hardware multiply/divide. In patient monitoring, if we are only interested in the simple averaging of the readings of parameters, the advantage of having hardware multiply/divide option may not be great as far as the amount of the saving of computer time is concerned because this type of operations is used at a rather low rate. However, if more complicated computation is required, such as the standard deviation as well as the mean of ECG, then the number of multiply/divide operations involved may be as many as 15 times the number of these operations involved in taking simply the average. In a real-time environment, such an increase of computer time required to perform these tasks may be quite expensive and a hardware multiply/divide option can be justified to be included in the system. It is noted that in calculating the amount of saving in computer time with the hardware multiply/divide option, we should include those tasks containing hidden multiply operations into consideration. For example, computing the address of an element in a 3-dimensional array involves two hidden multiply operations. The option of hardware floating point arithmetic may be evaluated in a similar manner. However, the computation with floating point arithmetic or some normalized integer algorithms can be used with low overhead. Furthermore, it should be noted that the addition of (integer) hardware multiply/divide enhances the performance of the software floating point arithmetic. Unless algorithms with a wide range of numbers are frequently used, it may not be worthwhile to include this option in the system.

Another important consideration is the trade-off between the size of the core memory and the use of a disk to store programs which are loaded as needed. With a priority assigned to each program, low priority routines can be suspended when higher priority routines are needed. Various methods for program relocation are available. However, they require that the programs are stored on the disk in an absolute format so that large amounts of computer time are not used up by a linking loader. The simplest relocation system is that when the system is generated, the programs are compiled or relocated into an absolute format which allows them to be run in only one core location. In this system, a lower priority routine using the core which is needed by a higher priority routine would have to be swapped out of the core as well as suspended. This requires another disk access which essentially doubles the time required to load the higher priority routine. Variations of this scheme allow the partitioning of any number of disjoint parts of the core memory in order to reduce the probability of the occurrence of a core requirement conflict and increase the probability that there will be some routine in the other parts of the core memory ready to run. This will prevent the computer from being idle during the swapping. Fig. 1 shows the amount of computer time lost to disk transfers of programs under various conditions. It is seen that there is essentially very little loss in computer time unless the number of disk transfers is large and the disk is slow (in Fig. 1 we use the following data: the moving head disk has an average access time of 60 milliseconds and the fixed head disk has only 16 milliseconds). Considering the high cost of the core memory (about $1 per word) and the relative low cost of the disk (normally less than 1 cent per word), these losses of computer time are probably justifiable even in the worst case. If we consider the use of a hardware relocation option, there are two different types. One is to use core mapping registers to make any page in the memory look like any other page. That is, by loading the relocation register, we can take any two free memory pages and make them look like pages 1 and 2 and load a two-page program which was relocated to run in pages 1 and 2. It may even be that the two pages are not adjacent or that page 1 is actually above (higher absolute address) page 2. This option can greatly reduce the probability that a swap will be needed and can sufficiently reduce the probability that the computer is going to be idle. The second type of relocation register is not as flexible as the first type, but allows us to set any location in memory to address 0 and the addresses above that location follow sequentially. Now, the core memory being used by any program must be a continuous block. Core relocation options are not available for many commercial minicomputer systems; however, their popularity is increasing.

Extra registers, both general purpose and index registers, are considered optional on some minicomputers. Index registers, although very useful to the programmer using assembly language, are seldomly supported by the high level language compilers, such as FORTRAN.

2) Addition of special purpose hardware.

Special purpose hardware can greatly increase the computing power of a patient monitoring system at a nominal cost. As an example, consider the patient monitoring system at the Chicago VA Hospital [6] which uses that data compression routine AZTEC developed by Cox [7,8]. This routine takes the digitized ECG and compresses it from the orthogonal 500 samples per patient per second. If this routine is used on 4 patients and requires about 100 microseconds per point, it will occupy 20% of the computer time Cox is currently developing a hardware preprocessor which would perform the AZTEC function. Since the algorithm is fairly simple, it is expected that the preprocessor will not be expensive, and yet can save about 20% computer time. In some areas, practically no commercial hardware is available to satisfy the specifications with a reasonable cost. One such area is the interactive multi-station display system. Under these circumstances, we have no alternative, but to design our own special purpose hardware to meet the requirements.

3) Software engineering

Proper use of the hardware, such as the index registers, can produce great savings in computing power. A recent study [9] indicated that up to 4 to 1 increase in computing power is obtainable by using assembly language over FORTRAN. Since this study was made on large-scale computers which have compilers producing every efficient object code, and since minicomputer compilers are somewhat simpler, it can be reasonably assumed that the saving of computing power by using assembly language will be larger on minicomputers than on large-scale computers. Software engineering can be used for this type of development, but it is fairly expensive especially if the number of systems to be produced is small. Thus, we should proceed with caution in this area. Much of the savings can be obtained at a low cost by using assembly language for the heavily used programs. A program using a large amount of computer time will often spend most of the time in a tight loop. In such a case, the loop should be written in assembly language as a subroutine. We should also consider modifying the software provided by the manufacturer. Since most software provided by the manufacturer are for general purpose use, we usually have to modify them in order to make them efficient. For instance, the real-time executive system available on a popular minicomputer system requires about 10K core memory and may not have all the desirable functions for our application. By dropping those unused functions and adding those desirable functions, it may be possible to reduce the core memory requirement of the real-time executive system from 10K to 6 to 7K.

AN EXAMPLE

In this section, we shall use the computerized patient monitoring system currently being designed at Northwestern University Biomedical Engineering Center as an example for illustrating some aspects of the above discussion. This system can be run in several different modes, but only two of these involving the use of a minicomputer as the central processor (CP) will be discussed here.

The Mode I system consists of the dotted line portion and deleting the preprocessor shown in Fig. 2. In this mode, the data is received from the bedside units directly through the analog-to-digital (A/D) converter. This produces an input rate of approximately 300 samples per patient per second, and hence puts a heavy load on the waveform recognition routines which becomes a severe limitation on how much this data could be stored. In this system, with 4 patients, the A/D conversion and waveform recognition routines will use about 40% of the CP time which is the largest single usage of the CP. In order to save CP time for this usage, we have the Mode II system, which does not use the dotted line portion shown in Fig. 2. In this mode, the CP receives its data from a preprocessor (PP). The PP is being built here as a special purpose piece of hardware, and is actually a small hybrid computer. Its function is to perform the equivalent waveform recognition outside the CP and then pass the results onto the CP. These results, which are the periods of the waveform, the width and amplitude of certain waveform components (e.g. the R-R interval, the QRS-width and R-peak of the ECG), and the sampled value of temperature, constitute about a 40 words/second input rate for all 4 patients. Consequently, we can use more sophisticated analysis routines than those in the Mode I system because more CP time can be allocated for them.

Since our patient monitoring system is expected to perform not only simple monitoring functions, but also somewhat complex analysis, we need a 16-bit minicomputer with 16K core memory, 1 microsecond memory cycle time, and a moving head disk. The block diagrams of the patient monitoring system and the minicomputer system are shown in Figs. 3 and 4. The minicomputer system is designed to be capable of shifting its operation from Mode I to Mode II and vice versa for realiability reason. The system is normally operated in Mode II; however, it will shift to Mode I operation if the PP fails. Of course, the complicated analysis routines will be temporarily suspended during the Mode I operation in order to monitor the same number of patients. On the other hand, in addition to its pre-processing purpose, the PP also provides certain kinds of monitoring functions when the CP fails. The plan we have for fault detection of the various portions of the system can be described as follows: In order to check the PP, we can set the CP to accept analog data occasionally, perform its own waveform recognition, and compare the results with the data being received from the PP. If there is a mismatch, then PP is operating improperly. The CP can also check the operation of the bedside units via special logic placed in them. However, so far there exists no commercial minicomputer system which can have its CP check itself. In order to accomplish the objective of checking the CP in real-time, we have developed a real-time fault detection scheme for minicomputers [10]. This scheme can be implemented by a simple hardware monitor added to the I/O bus of the computer like other I/O devices. This system should not require a full-time computer technician for its operation and should be capable of monitoring patients 24 hours a day. This system is designed to satisfy the simplicity-in-use requirement as stated before. Each bedside unit has, in addition to the normal transducers and their preamplifiers, a hybrid TV display for the PP, a CP generated TV display and a 16-button keyboard. The reason for selecting the TV display are low cost, familiarity to all the users, and ease in installation. The video display system for the CP generated display which also has some reliability features.

CONCLUSIONS

In this paper, we have discussed some aspects of the design of minicomputer systems for intensive care patient monitoring. It is obvious that many of the design considerations presented here are also applicable to other real-time systems. As new features are developed in minicomputers, the guidlines for systems design will change accordingly. One of the features which may affect the systems design greatly is the flexibility provided to the user in specifying the desired microprograms in the minicomputers for improving the performance on the given application.

ACKNOWLEDGMENTS

The authors would like to thank Professor J. E. Jacobs, Dr. M. H. Paul, M.D., Dr. H. U. Wessel, M.D., and Dr. R. L. Stout, all associated with the Biomedical Engineering Center of Northwestern University for helpful discussion and collaboration during the design and implementation of the computerized intensive care patient monitoring system.

REFERENCES

[1] H. R. Warner, R. M. Gardner and A. F. Toronto, "Computer-based monitoring of cardiovascular functions in postoperative patients." Circ. Suppl. II , Vol. 38, p. 68, 1968.

[2] D. W. Chaapel, G. Rastelli, and R. Wallace, "On-line computer care pf post operative cardiac patients," Digest of 1969 IEEE Computer Conf. , p. 105, Minneapolis.

[3] L. C. Sheppard, "Surgical intesive care process control system," Digest of 1969 IEEE Computer Conf., p. 113, Minneapolis.

[4] J. J. Osborn, J. O. Beaumong, J. C. A. Kaison, J. Russell, and F. Gerbode, "Measurement and monitoring of acutely ill patients by digital computer," Surgery, Vol. 16, p. 1057, 1968.

[5] F. J. Lewis, "Monitoring of patients in intensive care units," Surg. Clin. N. America, Vol 51, p. 15, 1971.

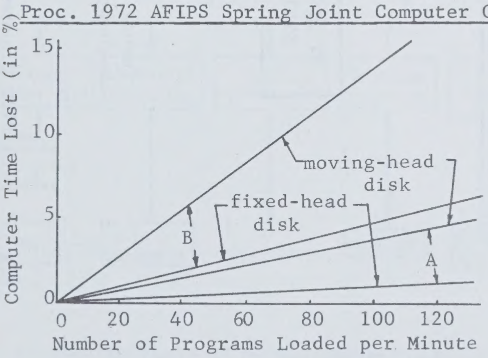
[6] H. Shubin, M. H. Weil, N. Palley, and A. A. Afifi, "Monitoring the critically ill patient with the aid of a digital computer," Computers and Biomedical Research, Vol. 4, p. 460, 1971.

[7] J. R. Cox, F. M. Nolle, H. A. Fozzard and G. C. Oliver, Jr., "AZTEC, a preprocessing program for real-time ECG rhythm analysis," IEEE Trans. on Biomedical Eng., Vol. 15, p. 128, 1968.

[8] J. R. Cox, F. M. Nolle, H. A. Fozzard, and G. C. Oliver, Jr., "Some data transformations useful in electrocardiography," Computers in Biomedical Research, Vol. 3, R. W. Stacy and D. D. Waxman, eds., New York: Academic Press 1969.

[9] J. O. Henriksen and R. E. Merwin, "Programming language efficiency in real-time software systems," Proc. 1972 AFIPS Spring Joint Computer Conf., Vol. 40, p. 155.

[10] J. R. Allen and S. S. Yau, "Real-time fault detection for saml1 computers," Proc. 1972 AFIPS Spring Joint Computer Conf., Vol. 40, p. 119.



A-For the case that the prob. for a swap being necessary and the prob. for the computer being idle during loading are both 25%.

B-For the case that the prob. for a swap being necessary and the prob. for the computer being idle during loading are both 75%.

Fig. 1. Computer Time Lost by Using Disk Resident Routines.

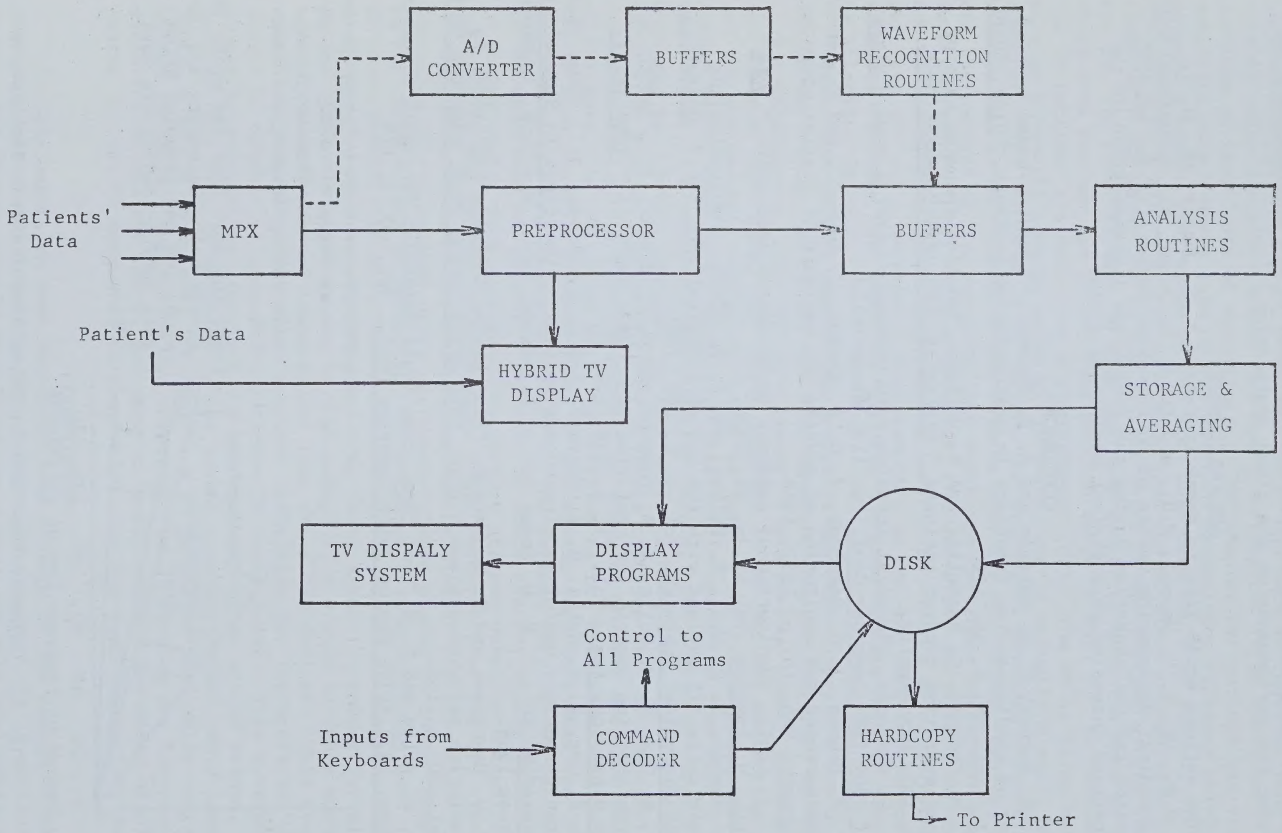

A/D CONVERTER
BUFFERS
WAVEFORM RECOGNITION ROUTINES
Patients' Data
MPX
PREPROCESSOR
BUFFERS
ANALYSIS ROUTINES
Patient's Data
HYBRID TV DISPLAY
STORAGE & AVERAGING
TV DISPALY SYSTEM
DISPLAY PROGRAMS
DISK
Control to All Programs
Inputs from Keyboards
COMMAND DECODER
HARDCOPY ROUTINES
To Printer

Fig. 2. Data Flow Diagram of Mode I and II of the Computerized Patient Monitoring System.

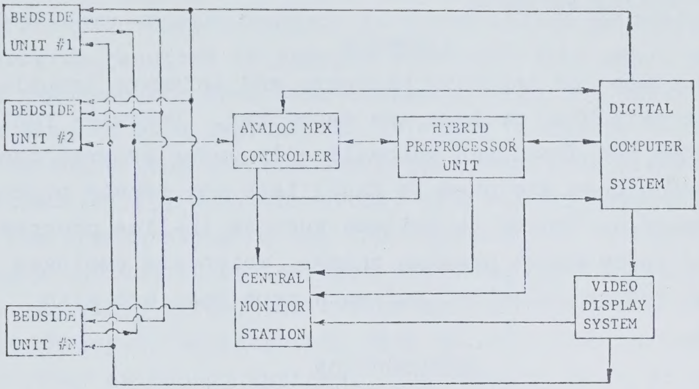

BEDSIDE UNIT #1
BEDSIDE UNIT #2
BEDSIDE UNIT #N
ANALOG MPX CONTROLLER
HYBRID PREPROCESSOR UNIT
DIGITAL COMPUTER SYSTEM
CENTRAL MONITOR STATION
VIDEO DISPLAY SYSTEM

Fig. 3. The Block Diagram of the Computerized Patient Monitoring System.

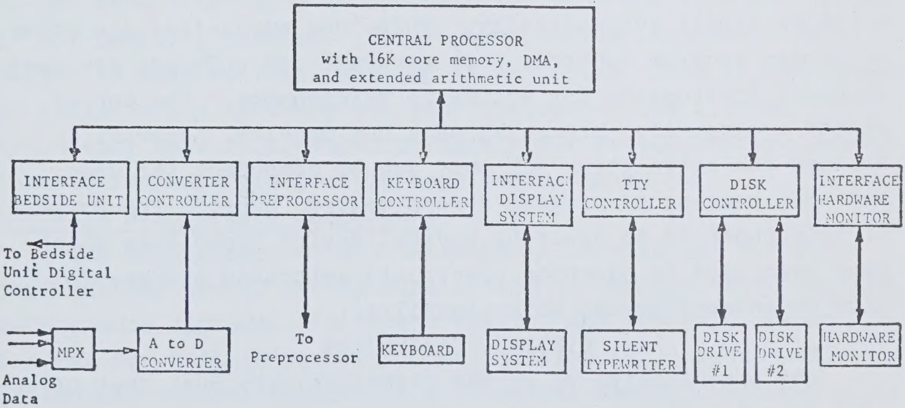

CENTRAL PROCESSOR with 16K core memory, DMA, and extended arithmetic unit
INTERFACE BEDSIDE UNIT
CONVERTER CONTROLLER
INTERFACE PREPROCESSOR
KEYBOARD CONTROLLER
INTERFACE DISPLAY SYSTEM
TTY CONTROLLER
DISK CONTROLLER
INTERFACE HARDWARE MONITOR
To Bedside Unit Digital Controller
MPX
Analog Data
A to D CONVERTER
To Preprocessor
KEYBOARD
DISPLAY SYSTEM
SILENT TYPEWRITER
DISK DRIVE #1
DISK DRIVE #2
HARDWARE MONITOR

Fig. 4. The Block Diagram of the Minicomputer System for the Computerized Patient Monitoring System.

HARDWARE MAINTAINABILITY AND SOFTWARE RELIABILITY OF ELECTRONIC SWITCHING SYSTEMS

HERBERT Y. CHANG
Bell Laboratories
Naperville, Illinois

ABSTRACT

Techniques for handling hardware and software troubles of electronic switching systems are described. Hardware faults are detected and diagnosed automatically under program control. Plug-in techniques are used to facilitate the repair process.

Software defensive techniques such as in-line program checks and independent program checks, which are employed to detect and reduce potential software problems, are also discussed.

INTRODUCTION

During the past several years, a number of electronic switching systems (ESS) have been developed, manufactured, and introduced in the Bell System to satisfy the variety of telephone switching needs [1]. One of the stringent requirements these systems must satisfy is the provision of continuous and accurate service 24 hours a day throughout the design life of the system. Thus, a high degree of dependability and maintainability has been incorporated into the system design.

A unique attribute in the philosophy of maintenance for ESS is that the machine itself can and should locate system troubles almost automatically. This idea emphasizes the power of stored program control; both hardware and software are used as major instruments for automatic maintenance. The use of stored program also brings along a new problem, however. Systems can malfunction not only due to component failures, but also because of program errors called "bugs." The purpose of this paper is to describe several design techniques which have been used in planning electronic switching systems for high dependability and maintainability.

SYSTEM ORGANIZATION

The organization of an ESS resembles very much that of a time-shared multiprogrammed computer operating in real time. The system utilizes the high speed of electronic devices to perform the basic switching functions of the logic, memory, and input/output control. The system actions are determined by a program stored in semipermanent memory; this enables one to provide and modify features by merely changing the stored program rather than by adding apparatus, wiring, and/or equipment options. System control is accomplished by subdividing the program required to process each call into small segments; these segments are then time-shared between the calls. A block diagram of a typical ESS is shown in Figure 1. Detailed description of the various subsystems have been previously described [2],[3] and, therefore, will not be repeated here.

The organization of the ESS hardware and software is strongly influenced by the fact that they must operate in real time. In other words, the system must at all times be able to accept an input, whether this data be submitted by customer, or by other switching systems. The failure to do so may result in improper handling of calls and, in general, degradation of service. Thus, the system must be able to quickly respond to any malfunctions (e.g., a component failure, a software bug, etc.) that may occur. This means that troubles must be quickly detected, isolated, identified, and removed. The techniques for achieving these objectives are discussed next.

MAINTENANCE PLAN

The use of stored program concepts and centralization of control functions have many advantages, but can also make the system more vulnerable to hardware failures and software errors if not properly handled. To preserve system integrity, duplication of some of the system units and other forms of redundancy in software and hardware are used throughout the system. Redundancy here is provided primarily for dependability, rather than to increase the traffic handling capacity.

Hardware Maintenance Techniques

The maintenance of ESS hardware is carried out through the combined use of system duplication to enhance dependability, maintenance circuits to detect faults, and the maintenance program for failure recovery and isolation [4]. Major units of central control complex, i.e., central control, program stores and call stores are duplicated. Similar redundancy is also provided in some peripheral units where a failure may disrupt the service to a large group of subscribers.

Specially designed maintenance circuits are added for fault detection purposes. Some examples are the match circuits for comparing the states of duplicated units, error detection/correction circuits associated with the memory units, 1-out-n-checks on commands, and many other internal waveform and timing checks. Additional access for diagnostic testing is also provided by way of test points at key locations within the units. These maintenance circuits have proven to be most useful in recognizing the presence of a failure quickly so as to initiate system recovery procedure.

Once the trouble is detected, the control is transferred from the call processing program to a set of maintenance programs called the fault recognition programs. The function of fault recognition programs is to quickly determine an operational configuration, establish it by appropriate switching of duplicating units, and then return to call processing. The objective here is to permit the system to recover from component failures rapidly and automatically, with minimal interruption of call processing. After faulty equipment has been put out of service, diagnostic programs are called to diagnose the trouble in the faulty equipment. The results of the diagnostic program are then printed out, specifying the location of the trouble. Plug-in techniques are used to facilitate the replacement of faulty circuits by the maintenance craftsmen.

Software Maintenance Techniques

There are two types of software defense techniques: the preventive technique and the after-the-fact technique. The preventive techniques are those specifically used in the design of program and data to detect software errors before they cause system misbehavior. Two of the most commonly used techniques in this category are the defensive programming method and the audits. The after-the-fact techniques are used to detect and/or correct software errors after they have occurred. This means that the system function has been either fully or partially affected and some malfunctions have been observed. Progressive system reinitialization is a very common after-the-fact approach and is used in many ESSs.

A. Preventive Techniques - Defensive programming techniques are used generally in the program design stage. Because these techniques are very much dependent upon the purposes for which the specific program is designed, and because programmers' skill and style also vary, a variety of methods have appeared. Some of the approaches are: (1) the range check to avoid erroneous transfers; (2) the state checks to ensure hardware state and memory state are in agreement; and (3) conditions checks on inputs to detect abnormal data.

Another type of defense which is designed primarily to perform logical checks using redundancy in the software system is called audits [5],[6]. Audits are a collection of independent check programs which detect and correct errors in memory content. In order to prevent the mutilation of data in memory, audits are run quite frequently and can be scheduled interleaved with other call processing tasks during the normal operations. Some of the audits also can run during system reinitialization.

Some of the techniques used to audit temporary memory data are: (1) the linkage checks on linked list structures; (2) consistency checks to verify constant data and parameters are intact; (3) timing checks to ascertain that no facility is being used beyond its maximum allowed timing limit; and (4) integrity checks to compare some redundantly stored data to detect state discrepancies of the facilities involved.

Upon detection of errors by audit programs, several courses of action can be taken. If the proper data is known, the erroneous information can be simply replaced by the correct information. Often, it is rather difficult to determine the exact source of error; the control will simply return to a "safe" (i.e., idle) state, if only a single call is affected.

When data mutilation is suspected which could gradually affect a large number of calls, the system control will call upon reinitialization or emergency action programs. These techniques are the subject of discussion for the next session.

B. After-The-Fact Techniques - When an error occurs that causes call processing capability to deteriorate drastically, the error may affect many parts of the facility. This situation sometimes may appear as a legitimate overload condition of the system. Thus, techniques for detecting such severe malfunctions are often very important. Since the base level program cycles repeatedly through the various call processing programs on a periodic rate, an effective way of checking the system status is by monitoring the program cycle time. A long cycle time, averaged over a number of consecutive cycles, is an indication of a system overload. At this point, an overload control limiting the rate at which inputs are accepted can be applied. If the overload control action turns out to be ineffective, then memory mutilation is a probable cause of the long cycle.

Once it is discovered that the error has degraded the system to the point that the call processing capability appeared to be severely hampered, all attempts to process calls are suspended and a system of emergency action programs is called upon to recover the system sanity. The emergency action programs usually consist of several consecutive "phases," each of which is automatically started if the condition is not corrected by the preceding phase. First, the reference data in the call store where transient information is stored are regenerated from office constants contained in the program store. Second, if the problem continues, certain portions of the call store are rebuilt. And finally, if necessary, a manually started reinitialization can be performed to initialize all the temporary memory to an idle state and to initialize all the circuits, including the switching network. This step is the most severe one. It terminates all the calls that are in progress at the time of the trouble; this, however, is designed to be executed very infrequently.

CONCLUDING REMARKS

More than several million hours of service have been logged on the ESS offices. Reliability and maintainability performance of both the hardware and the software have been good. The number of customer trouble reports, which is a measure of the quality of service used by operating companies, has been low.

Since the ability to provide high-quality, continuous telephone service is one of the prime objectives of a telephone switching system, improved maintenance techniques for both hardware and software should be continuously pursued.

REFERENCES

[1] Higgins, W. H. C., "A Decade of ESS," Bell Laboratories Record, Vol. 48, No. 11, pp. 318-325, December, 1970.

[2] "No. 1 Electronic Switching System," Bell System Technical Journal, Vol. 43, No. 5, Parts 1 and 2, September, 1964.

[3] "No. 2 Electronic Switching System," Bell System Technical Journal, Vol. 48, No. 8, October, 1969.

[4] Downing, R. W., et al., "No. 1 ESS Maintenance Plan," Bell System Technical Journal, Vol. 43, pp. 1961-2020, September, 1964.

[5] Nowak, J. S., et al., "Memory Mutilation in Stored Program Controlled Telephone System," 1970 International Conference of Communications, pp. 43-32-43-45.

[6] Connet, J. R., et al., "Software Defenses in Real Time Control Systems," Proceedings of 2nd Annual International Symposium on Fault Tolerant Computing, June 19-21, 1972, Boston, Massachusetts.

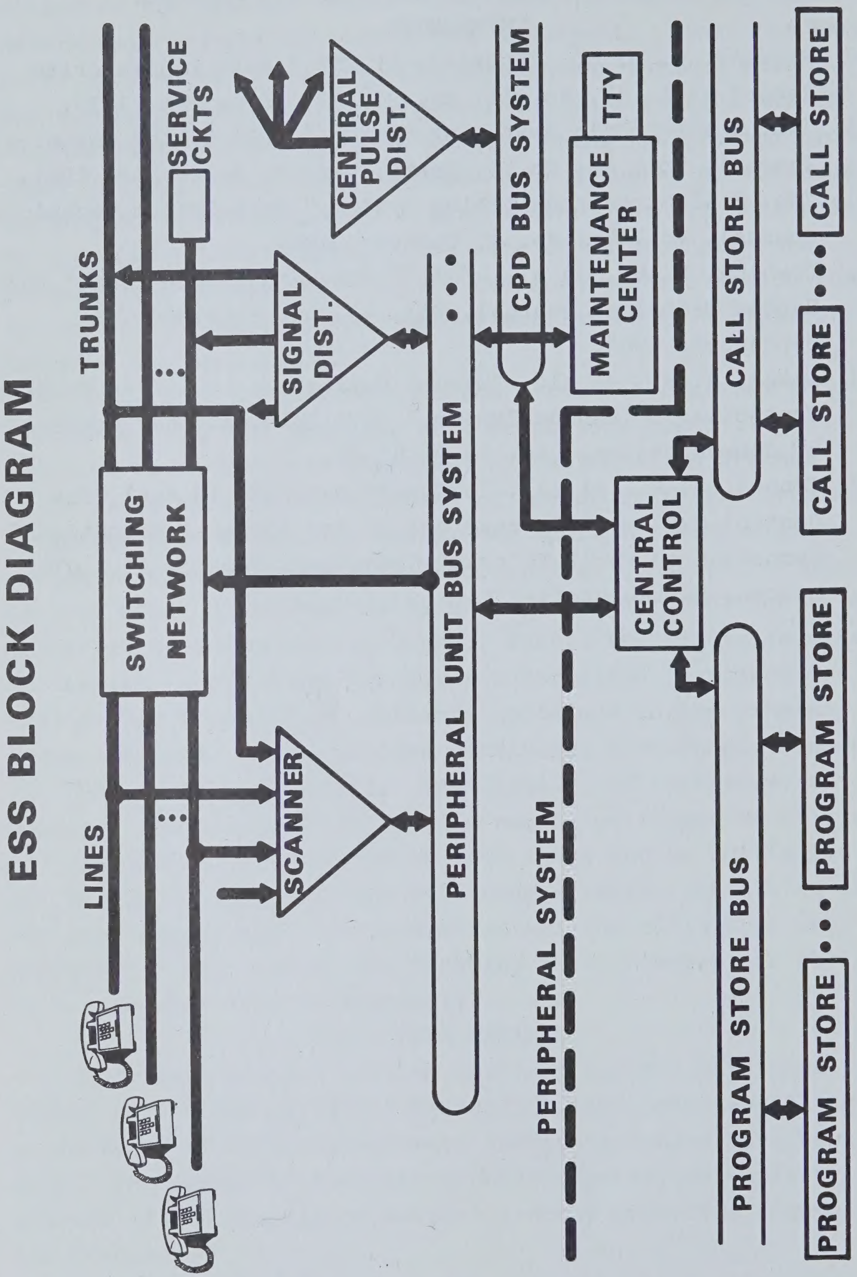

ESS BLOCK DIAGRAM
LINES
TRUNKS
SERVICE CKTS
SWITCHING NETWORK
SCANNER
SIGNAL DIST.
CENTRAL PULSE DIST.
PERIPHERAL UNIT BUS SYSTEM
CPD BUS SYSTEM
MAINTENANCE CENTER
TTY
PERIPHERAL SYSTEM
CENTRAL CONTROL
CALL STORE BUS
CALL STORE
CALL STORE
PROGRAM STORE BUS
PROGRAM STORE
PROGRAM STORE

Figure 1

TOWARDS A THEORY OF PIPELINED COMPUTING SYSTEMS

C.V. RAMAMOORTHY & S.S. REDDI
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, 94720.

ABSTRACT

This paper surveys the current status of pipelined Computer systems. A theory of pipelined Computer systems is proposed and the systems are classified and analyzed primarily from the point of view of job sequencing.

INTRODUCTION

The primary objectives in the computer architectures are to improve the turnaround times (total execution times of individual computational streams) and to increase the number of computational tasks performed per unit time (throughput rate) for a specified application spectrum and cost range. When computational rates beyond those achievable under current technology are required, parallelism within the problem structure must be exploited. This implies the need of languages that help indicate parallelism in the algorithms and methods of automatically recognizing the parallel processable segments in a sequentially organized computer programs. Also computer structures that are capable of parallel computation become necessary. The objective of improving the processing rate of computational tasks is sometimes synonymous with maximization of the utilization of resources of the system. This is achieved by sequencing computations through independent units of the system such that their utilization is enhanced. Both of the above objectives are feasible to implement since the modern LSI era can provide necessary hardware (i.e. various functional units capable of simultaneous and independent operation, multiplicity of the functional units, etc.) inexpensively.

Concurrent operation of system components can exist at different levels. At the highest level the operations of processors and I/O devices can be effectively overlapped in a multiple job stream to obtain efficient utilization of resources and increase in the throughput rate. Processor operations can also be overlapped with memory operations to improve the instruction execution rate. Multiple parallel processable instruction streams within the same job can be interleaved so that when one stream is using I/O devices, the other stream could be using the processor thereby improving the turnaround time for the job.

The concept of pipelining revolves around the concept of concurrent processing of independent tasks in a stream by independent individual units (such as I/O, processor, etc.) to improve the throughput rate. Basically, a pipeline may consist of work-stations (processing functions) through which jobs pass. Each work-station is capable of a certain function and whenever a job requires a function, the work-station specializing in that function starts processing that job provided that the work-station is free. (As can be seen we are assuming a nonpreemptive situation but preemption in pipeline environment is usually not preferable.) Since a job may consist of different tasks needing different processing functions, concurrent processing of a stream of jobs is possible due to the specialization of work-stations and their independence of each other. Consider the following situation which illustrates the pipeline concept[1]. Any instruction execution process can be functionally partitioned into the following subprocesses: instruction fetch, operation decode, development of operand address, operand fetch and operation execution. By dedicating special purpose functional device for each subprocess, instruction streams are passed through the pipeline one after the other. (See Fig. 1). The above pipeline can be termed linear since the jobs pass through in a fixed straight line order. In general, pipeline need not be linear and the jobs can have different processing orders.

Evolution of the Principle of Pipelining:

In the late 50's IBM 7090 and Honeywell 800 incorporated the first I/O and CPU overlapped operations between multiple job streams, a standard characteristic of all present computers. Multiprogrammed computers, featuring this characteristic, require more memory and I/O devices to be effective. They also require a control program to sequence the different programs and operate the asynchronous devices in the system. With an increase in the number of programs and devices, the complexity of the control program as well as the time spent in its execution grows. Recent attempts to overcome this have been to hardwire the critical and frequently used functions of the operating system (for example, SYMBOL computer) and use specially structured computers for specific problem types (e.g., ILLIAC IV, PEPE, etc.).

The overlapping of processor and memory operations is a feature found in some large scale machines like STRETCH. Effective use of this method can be helpful in improving the execution time of scientific computations like evaluation of arithmetic expressions. The method requires asynchronous control of memory and processors which implies hardwired implementation of control promitives for synchronization between the devices.

In the late sixties, the series of powerful computers introduced by IBM (360/91, 360/195, etc.) introduced pipelined execution of instructions. Since the instruction execution processing is almost fixed except for the final operation to be performed, its component subprocesses can be individually hardwired as work-stations and overlap between the subprocesses of successive instructions can be achieved. This process of pipelining achieved tremendous increase in the instruction execution rate, even in small computers like Super Nova. Where instruction execution process include subsidiary functions like instruction look-ahead, indexing, etc., they too are overlapped with other functions in the pipeline.

Current Theories and Techniques of Pipelining:

So far, there has been no satisfactory general theory of pipelining. Chen[2] proposes a theory based on geometrical and graphical concepts but unfortunately his theory, though gives some insight into pipelining, does not provide adequate basis for sequencing aspects that are essential for practical implementation of pipelining. Heller[3] proposes a theory considering sequencing aspects of multiprogrammed computer systems. He bases his theory on the concepts of machine shop scheduling. However, his theory has to be considerably modified to make it applicable to pipeline systems. The theory lacks depth and generality (which may be due to the fact that it was proposed in early sixties when machine shop scheduling was relatively less developed).

Let us examine briefly the existing techniques of pipelining in some of the computer systems. In IBM 360/91 the various subfunctions involved in instruction handling (such as instruction fetch, decode, etc.) are efficiently pipelined. The floating-point execution unit adopts the principle of common data bus[4] (CDB) to achieve overlapping in this unit. Reservation stations are provided for units like adders, multipliers, to facilitate enhanced job flow and tags are introduced to execute the various operations in an overlapped mode but at the same time preserving the precedence relationship between them. The following limitations can be observed in the CDB technique.

(1) The implementation relies heavily on the resources (because of tags, identification numbers, etc.) and hence, addition of any new resources cannot be easily incorporated.

(2) The system assumespseudo-register-to-register instructions.

(3) As pointed out in reference 4 performance is degraded when mixed precision arithmetic is involved.

(4) CDB becomes a bottleneck during heavy activity periods which may reduce the resource utilization of the computer system.

(5) Though helpful in increasing job flow, provision of reservation stations cannot always guarantee optimality.

(6) Degradation of performance in case of mixed arithmetic can be noticed because of routing restrictions.

(7) Programmer still has the burden of optimizing the code since different codes can give rise to different levels of concurrency in the system.

CDC 6600 employs[5] "function parallelism" and achieves the overlap of execution in the functional units. Central processor is effectively hidden behind central memory from the peripheral processors and the ten functional units (add, multiply, divide units, etc.), capable of independent and simultaneous operation, are hidden behind the central registers from central memory. This allows concurrency of operations in peripheral processors and central processor, and in the functional units. The instructions are central register-to-central register format. A "score-board" is provided to supervise concurrency and precedence of the tasks in execution and sequence the operations on the functional units properly. (The "score-board" scheme can be seen to correspond roughly to IBM's CDB technique). An instruction stack, capable of holding up to 32 instructions, is provided which may facilitate execution of loops by cutting down the memory accesses.

One can see CDC 6600 shares some of the merits and demerits of IBM 360/91. Though the "score-board" scheme is more flexible and accommodating than the CDB technique, it still performs local optimization and does virtually no sequencing operations. The congestion of "score-board" may reduce the efficiency of resource utilization.

The IBM STRETCH[6] computer system employs instruction and look ahead units to achieve pipelining in the stream of instructions. The instruction unit fetches the instructions, indexes and partially executes them and initiates memory references for the required operands. The look ahead unit consists of several look ahead levels. Each look ahead level is capable of holding a pre-decoded instruction from the instruction unit and receiving the operands required from the memory. As soon as an instruction is ready to be processed, the look ahead unit communicates with the arithmetic unit for processing the instruction. The look ahead unit also serves as a buffer to memory for store operations thus making it possible for the arithmetic unit to proceed to the next instruction.

STRETCH also features interesting characteristics such as forwarding, proceeding with preparation of instructions in cases of conditional branches assuming the branch will fail, interleaving of memories, etc. (In fact, most of the features can be found in later advanced systems like IBM 360/195, CDC 7600, etc.). One can see even a sophisticated system like STRETCH assumes the sequence of instructions as specified by the input program. Resource utilization may not be high and it is conceivable that instruction and look ahead units may sometimes lead to poor performances because of the strictly local optimization these units perform.

Proposed Theory of Pipelining:

Our proposed theory is more in the spirit of Heller. However, it is more profound and detailed than his theory. Our theory develops a strong basis on which pipeline sequencing and execution decisions can be made and motivates practical design of efficient pipeline computer system. It should be stressed that the theory does not guarantee efficient and optimal techniques for every pipelining situation because of either prohibitive nature of computations involved in generating these techniques and/or inadequacy of the theory to take into account all the features of the situation properly. Hence it is appropriate to point out that it would be desirable to develop heuristic techniques to tackle the above situations when the theory does not produce optimal techniques.

The pipeline concept is based on the idea of concurrent processing of independent tasks in a stream of jobs by independent functional units (FU's) to improve the throughput rate of the jobs. The definition of jobs and functional units vary according to the nature and level of the system we are interested in. For example when a multiprogrammed computer system is considered, the jobs may be individual programs and the FU's may be central processor, input/output and memory units. When a batch processing computer is considered, the jobs may be the different statements or subroutines of a single program and the FU's may be registers, arithmetic units, memory units, etc. Each job, in either case, may consist of tasks (partially or completely ordered by some given precedence constraints) require processing on the FU's and the throughput is enhanced by overlapping execution of independent tasks in the stream of jobs on the independent functional units. We talk in general terms such as jobs, functional units, etc.

A classification of pipeline systems is presented now. The classification is based on the kind of jobs to be executed and the environment of FU's. Let us define and explain the following terms and then proceed to the classification scheme.

Facility: Facility refers to a single functional unit or a group of identical functional units depending on the pipeline system. Any FU can process only one job at a time. When we say a job is to be processed by a facility, we mean the job is to be processed by any single FU belonging to the facility only once.

Facility Processing Order (FPO): The facility processing order refers to the specified order of facilities the job goes through for processing.

Schedule: A schedule of a given set of jobs is a scheme which specifies processing order of the jobs on the various facilities. All the schedules are assumed to be active, i.e., the jobs are started as soon as possible according to the processing order specified by the schedule. Sometimes schedules can be given by means of permutations in which case they are referred to as permutation schedules. The terms sequence and schedule are used interchangeably.

Intermediate Storage (IS): Intermediate storage is the storage needed to hold any partially processed job waiting for a busy facility (processing some other job). For example, consider jobs 1 and 2 with processing times 3 and 1 units on facility A, and 4 and 6 units on facility B respectively. Assume the jobs have to be processed first by A and then followed by B. Let the schedule (1,2) represent the processing sequence of job 1 followed job 2 on both facilities. Then for this schedule, one can see at 3 units of time job 1 finishes processing on A and starts processing on B until 7th unit of time. Job 2 starts processing on A and finishes at 4th unit of time, and finds B busy processing job 1. Hence job 2 has to wait until the time when B becomes available (in this case, until 7th unit of time). If the system has intermediate storage, then job 2 can be stored in this storage during its waiting period 4-7. Intermediate storage can be seen to be the storage used to save the partially proce-sed jobs that cannot be further processed because of either busy facilities or scheduling requirements. This storage should not be confused with speed-matching buffers provided between slow and fast devices. The capacity of IS refers to the maximum number of jobs the IS refers to the maximum number of jobs the IS can hold. We assume IS to be common to all waiting jobs. Now we give the following classification.

Classification:

(1) Jobs

(a) Nature of Jobs

FF: Flow-Facility. The jobs have the same FPO and the FPO does not contain repetition of any facility, i.e., any job processes on any facility only once.

JF: Job-Facility. The jobs do not have a common FPO and the FPO for any job may contain repetition of any facility.

(Sometimes we refer to FF and JF systems as uni-flow and bi-flow pipeline systems. For if one imagines a pipeline to consist of all the facilities, then for FF the jobs can be seen to flow unidirectionally whereas in JF the flow of jobs may not be unidirectional).

(b) Ordering of Jobs

JI: Independent jobs. All the jobs are independent of each other.

JD: Dependent jobs. Some jobs depend on some other jobs which means that some jobs can be initiated only when some other jobs are finished processing. The dependencies may be specified by a partial order of jobs.

(2) Environment of Facilities

(a) Intermediate Storage

NIS: No Intermediate Storage. The capacity of IS is zero.

FIS: Finite Intermediate Storage. The capacity of IS is finite but not zero.

IIS: Infinite Intermediate Storage. The capacity of IS is infinite for all practical purposes.

Now a system characterized by FF/JD/FIS refers to uni-flow pipeline system with finite intermediate storage and job dependencies. By FF/IIS we mean the systems FF/JI/IIS and FF/JD/IIS.

To study and analyze these systems one should make certain assumptions regarding the nature of the processing times, i.e., whether the processing times can be predicted deterministically or probabilistically etc., and decide the desired objectives, i.e., whether to minimize the completion time of all the jobs, etc. The objective function of minimizing the completion time of all the jobs (which is equivalent to maximization of the throughput rate) seems to be reasonable because firstly, in most of the computer systems the throughput rate essentially decides the performance of the system and secondly, this objective seems to be more amenable to mathematical analysis and hence investigating with this objective may lead to understanding and possibly some heuristic techniques in solving the systems with other objectives. FF/IIS and JF/IIS systems have been investigated extensively in the field of operations research[7]. In reference 8, results are presented for some of the pipeline systems under the assumption of deterministic processing times. Because of space limitations we do not go into the details regarding the techniques developed for these systems.

Proposed Pipeline System:

Now a simple pipeline system is proposed. Unlike the existing techniques where most of the sequencing is specified by the programmer, the proposed system has a separate unit, called Sequencer, which analyzes the input code and decides the sequences of the various tasks of the program on the various facilities. The Sequencer may consist of a recognizer as described in reference 9 and a scheduler. The recognizer accepts the input code and generates information regarding the dependence relationships between the various tasks of the input program. The scheduler examines the task graphs of the program exhibiting the dependencies of the tasks, and by means of the techniques developed for the pipeline system, produces schedules of the tasks on the facilities. Then the Sequencer communicates this scheduling information to the Execution unit (this unit is essentially a conventional CPU) which processes the tasks of the program on the facilities according to the schedules specified by the Sequencer. The Sequencer and the Executor are separate individual units and are capable of simultaneous operation. Fig. 2 shows diagrammatically the whole system. The time spent by the Sequencer in analyzing the input and generating the schedules is referred to as overhead and the time of computation on the Executor is called execution time. One can see that it is possible to segment a program and overlap overhead and execution times to improve the performance of the system.

Depending on the way the Sequencer and the Executor interact, we have two models, Static Decision Model (SDM) and Dynamic Decision Model (DDM). These models can be explained by an example. Consider a program whose task graph consist of 100 tasks. One has the option regarding the segmentation of the program. He can execute the whole program together or segment the program into two subprograms each containing 50 tasks. In the later case, one can analyze the 50 tasks of the first subprogram to get a schedule, execute this subprogram on the Executor and at the same time start analyzing the second subprogram on the Sequencer, and finally executing the second subprogram on the Executor. (It can be seen that the first subprogram should be independent of the second one for this scheme to work). A decision has to be made regarding the number of the tasks (the size of segmentation) to be analyzed at one time (the segmentation size is also referred to as partition size). In SDM, the partition size is fixed (say at P tasks) and the program is segmented in sets of P tasks (probably the final set may not contain exactly P tasks), analyzed and executed in an overlap mode. In DDM, the partition size is variable and depends on the execution times of the subprograms. It starts with the first subprogram containing say 10 or 15 tasks, analyzes and executes the subprogram. Then based on an estimate of the execution time of the first subprogram, decision is made regarding the partition size of the next subprogram. This procedure is continued until the completion of the program. It is apparent that for the model to work one should have information regarding the dependence of overhead on partition size. Now each model is considered separately and the factors that decide the performance are presented.

Static Decision Model: It is assumed that our primary concern in analyzing these systems is to maximize the throughput rate of the jobs. One of the parameters deciding the performance of the system is the partition size. The effect of partition size on the completion time of any given program can be investigated by simulation and based on this we can choose the partition size that gives optimum performance. The optimum partition size depends on the processing times of the tasks to be executed, the overhead involved in generating the schedules and the structure of the dependency task graph. One can note that with larger partition size overhead will be more but at the same time, execution time will be less because of the increased overlap between the tasks. With smaller partition size overhead will be less but at the expense of more execution time because of the decreased overlap. Hence a 'good' partition size compromises between overhead and execution time. When the processing times of the tasks are extremely small compared with overhead times, pipelining may not promise any gains in the performance of the system. On the contrary it may be better off not to use pipelining under these circumstances.

The way the tasks of the input program are numbered has an effect on the system performance. If one considers the whole program without segmentation, it is obvious that the system performance is independent of the numbering of the tasks. But when segmentation is considered, the improvements in the throughput rate of the system depend on the structure of the task graph and the numbering scheme. Of course, an 'ideal' partition size exhibits minimum variance in system performance for different numberings. For this way the programmer is relieved the burden of optimally numbering the tasks, i.e., optimally arranging the statements of his program.

Dynamic Decision Model: The DDM is useful if variable partition size is allowed. Unlike SDM, the partition size is decided based on the processing times of the tasks in this model. Because of this, DDM avoids the situations where pipelining may degrade the performance of the system (which may happen due to the small processing times of the tasks and large overheads incurred in scheduling the tasks). DDM requires information regarding the partition size overhead dependence for its operation. The performance of the system depends on the initial partition size chosen and as in SDM, it also depends on the numbering of tasks and the structure of the task graph.

Simulation results of SDM are shown in Fig. 3. For the purposes of simulation it is assumed that the input program is in the form of task graphs. A heuristic based on Johnson's algorithm is used to schedule the tasks of the program (for details of the heuristic, see reference 8). For a given task graph, processing times are randomly generated for the various tasks and a scaling factor is attached to these processing times. By varying k we can obtain the various ranges for the processing times. Overhead time for a given partition of the program is assumed to be given by the CPU time of CDC 6600 (on which the simulation results are conducted) to generate a schedule, using the heuristic, for the partition. In the graph of Fig. 3a gain or loss refers to the percentage gain or loss of the completion time for the program of a pipelined system over a nonpipelined system. This graph shows the variation in gains with partition size and scaling factor (the input program is assumed to consist of 32 tasks). The graph of Fig. 3b shows the variance in gains with different numberings of nodes for different k against partition size. For partition size of 32 tasks (when the whole program is considered without segmentation) the variance should be ideally zero; but our heuristic relies sometimes on the numbering of the tasks for generating schedules and hence one can observe the nonzero variance.

Now let us point out advantages of the above proposed pipeline system. It possesses flexibility in the sense that new facilities can be easily added and the faulty facilities can be dropped without any problems. Efficient resource allocation is achieved by avoiding local optimization techniques and by having a sequencer instead of the programmer schedule the tasks. The restriction that the input code should be in a particular form can be relaxed.

Branch type of instructions (like computed and uncomputed GO TO statements, logical and arithmetic IF statements, etc., in FORTRAN language) pose problems in implementation of pipelining and in general, reduce the efficiency of pipeline systems. Branches can be definitive sometimes (as in unconditional GO TO statements) and these cases can be handled without much difficulty. However, conditional branches cannot be predicted until the conditions are computed. Let us see how we can tackle the branching situation for our proposed system. When an instruction like GO TO (30,40, 50)i is encountered, the Sequencer can check whether the value of i is available. If i is available, the branch becomes definitive and the Sequencer can proceed along the branch specified. If i is not available on the contrary, we have many options. We can halt all the computations until i becomes available or we can take an arbitrary path and proceed processing instructions along this chosen path. Clearly, the former case is not efficient. The latter option (used by computers like IBM, UNIVAC, etc.) involves retraction of paths if the chosen path proves to be wrong. In this option the Sequencer keeps track of the path taken, the memory locations and registers affected by choosing that path, etc., and examines its decision once i becomes available. If the chosen path proves to be wrong, the Sequencer issues commands to restore the contents of the affected memory locations and registers and to retract the path. If the chosen path proves to be right, no action is necessary and the Sequencer proceeds without any retraction. The Sequencer can be made capable of computing simple arithmetic and logical conditions. This way the Sequencer does not have to wait for the Executor to compute the conditions thereby using considerable time. For example, if there is a statement IF (A-B.LE.5.0) GO TO 30, the Sequencer can take an immediate branching decision once the values of A and B are known.

Since in our system the Sequencer makes the branching decisions, one can incorporate learning and adaptive features into the system easily. We can make the Sequencer keep track of the past history on branchings and decide the branchings based on this history. Programmer can specify to the Sequencer his preferences on branchings. Thus one can see the flexibility and superiority of the branching aspects of our system.

Conclusion:

Pipelining may be viewed as a step toward the goal of efficient resource utilization in the computer system. One should check whether the increase in the speed due to efficient resource allocation will pay for the investment made in incorporating pipelining in the computer architecture. Pipelining is an extremely useful concept for frequently occurring tasks. For these cases execution of the tasks can be pipelined (note the usefulness of pipelining is realized only if the tasks follow each other so that overlap can be achieved between the tasks). An example of the above situation is the instruction processing pipeline. In some computer systems, probably specialized, matrix operations may be frequent and in these cases the pipelining can speed up the performance considerably. The advantage in these cases lies in the fact that the designer knows beforehand the nature of the tasks, and his design can efficiently pipeline the processing equipment (probably by hardwiring) thus cutting down all the overhead that is needed in the absence of such knowledge of the nature of the tasks.

With large scale integration technology, the feasibility of inexpensive production of the required functional units can be reasonably expected. The questions one may face concern efficiency and economy of pipeline systems. Because of LSI era the economic considerations might not pose a severe problem. As for the efficiency the designer has to investigate the improvement in the performance due to pipelining. The performance can be improved by overlapping the overhead involved in pipeline sequencing decisions with the execution time. But it should be noted that performance is sensitive to problem structure. This suggests measures for evaluating feasibility and suitability of pipelining to solving a given problem. Though such measures have been investigated in connection with parallel systems[10], they appear to be complex to formulate for a pipeline system. Experience and simulation are good guides in assessing the suitability of pipelining to a particular problem.

Parallel and pipeline systems can be (and have been[11]) compared with respect to several criteria like cost, performance, performance/cost, performance/weight and volume of the hardware, etc. Parallel systems depend on replication of processing elements which increases cost of production and weight of the hardware. However, the principle of replication on the other hand makes production of equipment easier and facilitates convenient incorporation of fault diagnostics. Pipeline systems specialize the functional units and increase speed by concurrent operation of these units. Because of this they may not enjoy the advantages of easy production and incorporation of diagnostics as in parallel systems. But for the pipeline systems the hardware may not be extensive in comparison with replication systems. This feature makes pipelining desirable in the situations where high performance is demanded with less hardware. Hence airborne and space computing equipment may find pipelining an attractive principle. Interrupt handling and fault tolerance on the other hand may tend to complicate pipelining. On the whole the best approach seems to be a compromise between pipelining and replication, and the future trend will be toward this direction.

Before any pipeline system is designed and installed, it is important to test suitability and adaptability of the system to the environment in which it is going to be used. The system should possess flexibility and be capable of expansion. Evaluation of the system capabilities and efficiency is an important topic left for further investigations. Testing plays an important role after installation of the system. Fault tolerance, detection and diagnostics are desirable aspects in the pipeline systems and much work has to be done in this direction.

References:

[1] Anderson, D.W., et al.: "The IBM System/360 Model 91: Machine Philosophy and Instruction Handling," IBM j. of Res. and Dev., Vol. 11, No. 1, Jan. 1967.

[2] Chen, T.C.: "Parallelism, Pipelining and Computer Efficiency," Computer Design, Jan. 1971.

[3] Heller, J.: "Sequencing Aspects of Multiprogramming," J. of ACM, Vol. 8, No. 3, 1961.

[4] Tomasulo, J.E.: "An Efficient Algorithm for Exploiting Multiple Arithmetic Units," IBM J. of Res. and Dev., Vol. 11, No. 1, Jan. 1967.

[5] Thornton, J.E.: "Parallel Operation in the Control Data 6600," AFIPS Proc. FJCC, Pt. II, Vol. 26, 1964.

[6] Buchholz, W., (ed.): Planning a Computer System, McGraw-Hill Book Company, New York, 1962.

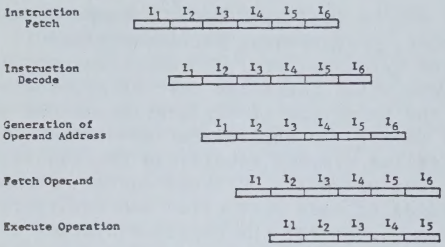
[7] Conway, R.W., et al.: Theory of Scheduling, Addison-Wesley, Reading, Mass., 1967.

[8] Reddi, S.S.: "Sequencing Strategies in Pipeline Computer Systems," Doctoral Dissertation, Department of Electrical Engineering, The University of Texas at Austin, Texas, Aug. 1972.

[9] "The FORTRAN Parallel Task Recognizer," Final NASA Rept., Grant NGR 44-012-144, May 1970.

[10] Gonzalez, M.J. and Ramamoorthy, C.V.: Program Suitability for Parallel Processing," IEEE Trans. Computers, Vol. C-20, No. 6, June 1971.

[11] Graham, W.R.: "The Parallel and the Pipeline Computers," Datametion, Vol. 16, April 1970.


Instruction Fetch
I_1 I_2 I_3 I_4 I_5 I_6
Instruction Decode
I_1 I_2 I_3 I_4 I_5 I_6
Generation of Operand Address
I_1 I_2 I_3 I_4 I_5 I_6
Fetch Operand
I_1 I_2 I_3 I_4 I_5 I_6
Execute Operation
I_1 I_2 I_3 I_4 I_5

An example of pipelining showing instruction process and execution. I_j represents process of jth instruction.

Fig. 1

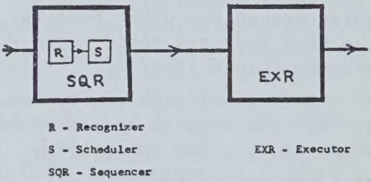
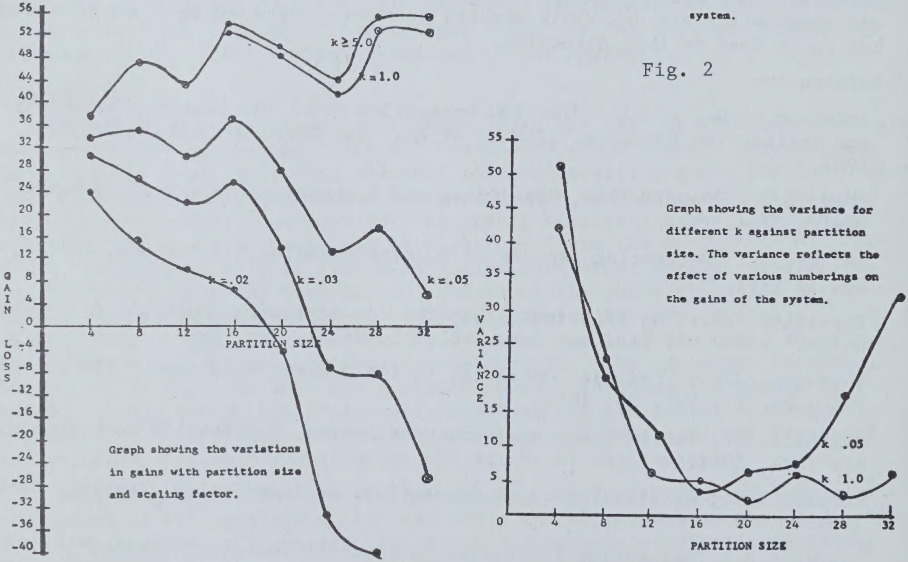

R
S
SQR
EXR
R - Recognizer
S - Scheduler
SQR - Sequencer
EXR - Executor

Diagram showing the proposed simple pipeline system.

Fig. 2



Graph showing the variations in gains with partition size and scaling factor.

(a)

Graph showing the variance for different k against partition size. The variance reflects the effect of various numberings on the gains of the system.

(b)

Fig. 3

AN APPLICATION OF IMPLICIT FUNCTION THEORY FOR PARAMETER IDENTIFICATION

J. R. CANNON
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

When two parameters are to be determined in a system, the implicit function theorem offers a possible method of determining these parameters from two intersecting graphs. A brief discussion of the general method will be given. Some examples will be discussed.

GENERAL METHOD

Consider a physical process, which we shall denote by T, which is describable in terms of some mathematical model. We suppose that the description of the process T depends upon a collection of parameters. In this description of T we shall assume that two of the parameters, say k_1 and k_2, are unknown. For example, in the case that T is the flow of heat in a nice conductor, we have the equation for the temperature u

$$k_1 \frac{\partial u}{\partial t} = k_2 \Delta u \tag{1}$$

as part of the description with k_1 as the product of the density and heat capacity of the conductor and k_2 as the conductivity. Now, let f denote input data for the process T and let v denote the output or the effect of the process T on f; i.e.,

$$v = Tf \ . \tag{2}$$

We remark here that v may depend upon many parameters, but for our purpose here it suffices to display only the dependence upon k_1, k_2, and f. Thus,

$$v = v(k_1,k_2;f) \ . \tag{3}$$

Suppose, now that $m = m(v)$ is a real valued functional defined over the set of possible outputs v that can result from all possible inputs f. Clearly,

$$\begin{aligned} m &= m(v(k_1,k_2;f)) \\ &= M(k_1,k_2;f) \end{aligned} \tag{4}$$

represents an experimental observation on the output v of the system. In the happy circumstance that

$$\frac{\partial M}{\partial k_1} \neq 0 , \tag{5}$$

it can follow from the implicit function theorem that (4) defines a functional relationship between k_1 and k_2:

$$k_1 = k_1(k_2,m;f) , \tag{6}$$

where

$$m = M(k_1(k_2,m;f),k_2;f) \tag{7}$$

serves as the defining relation for the function in (6). We remark that (5) can hold even if we are not able to demonstrate it with a rigorous mathematical argument. Under our basic assumption that the physical process T is describable by a mathematical model, it follows that for each and every choice of input data f, the true physical values of k_1 and k_2 must lie on each graph of k_1 versus k_2 as described by (6) and (7). Consequently, if the graphs vary with changes in input data, the true physical values of k_1 and k_2 can be obtained from the intersection point of two or more graphs of k_1 versus k_2. Indeed, the graphs of k_1 versus k_2 are easily constructed since (5) implies that M is either monotone increasing or decreasing in k_1 for fixed k_2 and f. For fixed f select a value of k_2. A simple search using the monotonicity of M with respect to k_1 will lead to a k_1^* and a k_1^{**} such that

$$M(k_1^*,k_2;f) \leq m < M(k_1^{**},k_2;f) . \tag{8}$$

If equality holds in (8), then

$$k_1(k_2,m;f) = k_1^* . \tag{9}$$

Otherwise, $k_1(k_2,m;f)$ can be approximated by the technique of interval halving applied to the interval (k_1^*,k_1^{**}). One only need select enough k_2 to adequately depict the graph.

EXAMPLE 1.

In [2], the author, D. L. Filmer, and N. Reiss considered the general chemical reaction

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} B \xrightarrow{k_2} C + X \ ,$$

$$C \xrightarrow{k_3} E + D \ ,$$

where k_1, k_{-1}, k_2, and k_3 are the kinetic rate constants. Since the reaction

$$C \xrightarrow{k_3} E + D$$

can be observed independently of the other, we assumed that k_3 was known. The mathematical model assumed for the reaction was

$$\begin{aligned} \dot{E} &= -k_1ES + k_{-1}B + k_3C \ , \\ \dot{S} &= -k_1ES + k_{-1}B \ , \\ \dot{B} &= k_1ES - k_{-1}B - k_2B \ , \\ \dot{X} &= k_2B \ , \\ \dot{C} &= k_2B - k_3C \ , \\ \dot{D} &= k_3C \ , \end{aligned} \tag{10}$$

where

$$\begin{aligned} E(0) &= E_0, \quad S(0) = S_0, \\ B(0) &= X(0) = C(0) = D(0) = 0 \end{aligned} \tag{11}$$

and where E_0 and S_0 are the positive initial concentrations. For enzyme reactions k_1 and k_{-1} are quite large and $k_{-1} \gg k_2 \gg k_3$. A general problem confronting the experimentalist is the determination of the constants K_m and k_2,

$$K_m = \frac{k_{-1} + k_2}{k_1} \doteq \frac{k_{-1}}{k_1} \ , \tag{12}$$

given only the initial conditions and a portion of the curve $X = X(t)$ which can be obtained from a color change of the solution. Taking

$$m = X(t_i, E_0, S_0, k_1, k_{-1}, k_2, k_3) \tag{13}$$

it was shown that for k_1 and k_{-1} large, the relation (13) defined a linear relation between K_m and k_2. It was shown that when E_0 and S_0 are changed, the linear relation is changed. Thus, the correct physical pair can be obtained from a pair of intersecting lines. Numerical experiments were conducted to illustrate the accuracy of the technique in the presence of measurement errors.

EXAMPLE 2.

In [3], the author, Paul C. DuChateau, and D. L. Filmer considered the chemical reaction

$$A + B \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \tag{14}$$

and the problem of determining the unknown rate constants k_1 and k_{-1} in the presence of diffusion. We have assumed that the physical process was governed by the following equations for $0 < x < L$, $t > 0$:

$$\begin{aligned} \frac{\partial A}{\partial t} &= k_A \frac{\partial^2 A}{\partial x^2} - k_1 AB + k_{-1} C , \\ \frac{\partial B}{\partial t} &= k_B \frac{\partial^2 B}{\partial x^2} - k_1 AB + k_{-1} C , \\ \frac{\partial C}{\partial t} &= k_C \frac{\partial^2 C}{\partial x^2} + k_1 AB - k_{-1} C , \end{aligned} \tag{15}$$

where $A = A(x,t)$, $B = B(x,t)$, $C = C(x,t)$ denote the concentrations of the chemicals A, B, C respectively, and k_A, k_B, and k_C denote the respective diffusion rates. The model was completed by requiring that

$$\frac{\partial A}{\partial x} = \frac{\partial B}{\partial x} = \frac{\partial C}{\partial x} = 0 \quad \text{at} \quad x = 0, L, t > 0 \tag{16}$$

and giving the initial configuration

$$\begin{aligned} A(x,0) &= \begin{cases} A_0 , & 0 \le x \le \theta L , \quad 0 < \theta < 1 , \\ 0 , & \theta L < x \le L , \end{cases} \\ B(x,0) &= \begin{cases} 0 , & 0 \le x \le \theta L , \\ B_0 , & \theta L < x \le L , \end{cases} \\ C(x,0) &= 0 , \quad 0 \le x \le L , \end{aligned} \tag{17}$$

where A_0 and B_0 are positive constants. For the additional information needed to determine k_1 and k_{-1} we assumed that we could obtain the measurement

$$m = \int_{x_1}^{x_2} C(x, t_*, k_1, k_{-1}, A_0, B_0)\,dx \tag{18}$$

for some $t_* > 0$. It was shown that (18) defined a functional relationship between k_1 and k_{-1}. Numerical experiments were conducted to demonstrate the feasibility of obtaining the constants k_1 and k_{-1} from intersecting graphs and to test the effects of errors in the given data.

EXAMPLE 3.

In [1], B. L. Buzbee and F. W. Dorr applied the technique to the determination of the thermal conductivity of graphite from experimental data. They demonstrated numerically that for very accurate data the technique would yield the thermal conductivity which was assumed to be in the form

$$k(u) = \frac{A}{u} + B \ , \tag{19}$$

where u is temperature and A and B are the parameters to be determined. When they applied the technique to the experimental data, they obtained three "parallel" lines; i.e., no intersection point. They concluded that either the data was too inaccurate or that the model was incorrectly chosen.

EXAMPLE 4.

Recently [4], the author and Paul C. DuChateau have considered the feasibility of applying the technique to the problem of determining the conductivity and the product of the density and the heat capacity of heat conductors.

BIBLIOGRAPHY

1. B. L. Buzbee and F. W. Dorr, A method for the numerical determination of thermal conductivity, Los Alamos Report LA-4485, September, 1970.

2. J. R. Cannon, D. L. Filmer, and N. Reiss, A method for determination of rate constants in enzyme reactions, J. Theoret. Biol. 16 (1967) 280-293.

3. J. R. Cannon, Paul C. DuChateau, and D. L. Filmer, A method of determining unknown rate constants in a chemical reaction in the presence of diffusion, Math. Bio. Sci. 9 (1970) 61-70.

4. J. R. Cannon and Paul C. DuChateau, Determination of unknown physical properties in heat conduction problems, submitted.

A DISTRIBUTED REPRESENTATION OF UPPER TORSO SAGITAL PLANE BENDING RESPONSE: MODELING AND PARAMETER IDENTIFICATION

L. DANIEL METZ, PH.D.
Department of General Engineering
University of Illinois at Urbana-Champaign

ABSTRACT

Upper torso bending response has traditionally been modeled as a lumped parameter system of interconnected rigid bodies. Such an obviously simplified model avoids many problems inherent in continuum representations of the living human torso at the expense of realism. This paper discusses modeling and parameter identification of the torso using a distributed-parameter approach.

INTRODUCTION

In the examination of the problem of sagital plane bending response, two approaches are possible. One approach is experimental, and involves the use of animals anthropologically similar to man (e.g., chimpanzees), dummies and, in some cases, cadavers in controlled experimental conditions [1,2]* A second approach is theoretical in nature, and uses modeling to generate a mathematical conceptualization of bending response.

Theoretical models of human dynamic response may be either lumped parameter or distributed parameter (continuous). Many examples of lumped parameter models exist in the literature [1,3]. These models characterize the human torso as a series of mass elements interconnected via pins, torsional springs and the like. The approach is difficult to apply to areas of the body where large, continuous masses can be appreciably deformed (for example the torso area), but works well for clearly defined "links" such as arms and legs.

Distributed parameter system models have also been postulated [4]. These represent the torso as a continuous, elastic beam, usually with uniform elastic and geometric properties, and describe torso time domain response in terms of such a beam model's response. This response is idealized due to the fact that classical theory of vibrations of continuous beams is based on some rather restrictive assumptions. Deflections are assumed to be small compared with the dimensions of the beam, and are (usually) also assumed to be inextensional, and shear force and rotatory inertia are often neglected. The theory is well developed however [5], and shear force and rotatory inertia are sometimes superimposed to improve the classical linear results [6,7].

Liu [4] used this classical, small motion theory to prescribe a model of the human torso and also attempted to include the effects of head inertia by placing a lumped mass at one end of the beam. He concluded (correctly) that the restriction of small dynamical deflections would preclude meaningful results for the problem of interest, and abandoned his distributed parameter model in favor of a lumped parameter one. Clearly, it is the presence of large geometric deflections which characterizes traumatic torso dynamic behavior; any model restricted to small transverse displacements is thus only of minimal value. However, a continuum model of torso behavior is intuitively more attractive than a lumped parameter one because of the continuously distributed physical properties of the torso itself.

GOVERNING DIFFERENTIAL EQUATION OF MOTION

Consider the development of the differential equation of motion for the large geometry, transverse vibrations of a continuous beam with a lumped mass at one end. Shearing and rotatory inertia are neglected, the beam is undergoing inextensional motion and has one end clamped and the other end free. The lumped mass at the end of the beam is taken to represent the head mass of a human, after the approach of Liu [4]. See Figures 1-3.

---

*Numbers in brackets designate references at the end of the paper.

Neglecting damping, the system is conservative and the technique of Hamilton's principle can be used to derive the governing differential equation of motion. The lumped mass is most easily treated by including its effects as boundary conditions applied at the free end of the vibrating beam. See Figure 4. The treatment of the derivation of the differential equation is thus parallel to that of Wagner [8].

The kinetic energy T of a beam of length ℓ can be written as:

$$T = \frac{\rho A \ell}{2} \int_0^1 (\dot{x}^2 + \dot{y}^2)\, d\xi \tag{1}$$

where a dot indicates differentiation with respect to time t, ρ = density of the beam material and $\xi = s/\ell$, the dimensionless arc length. The potential (strain) energy stored in the beam due to bending is:

$$V = \frac{EI\ell}{2} \int_0^1 K^2(\xi,t)\, d\xi \tag{2}$$

in which $K(\xi,t)$ is the curvature of the centerline of the beam and EI the flexural stiffness. From calculus, K can be expressed parametrically in terms of x and y:

$$K(\xi,t) = \lambda^3(x'y'' - x''y') \tag{3}$$

where a prime denotes differentiation with respect to dimensionless arc length ξ and $\lambda = 1/\ell$. Hamilton's principle requires that the integral:

$$I = \int_{t_1}^{t_2} (T - V)\, dt \tag{4}$$

have an extremum value, i.e., its first variation is zero. Expansion of (4) yields:

$$I = \int_{t_1}^{t_2} \int_0^1 [\rho A(\dot{x}^2 + \dot{y}^2) - EI\lambda^6(x'y'' - x''y')^2]\, d\xi dt \tag{5}$$

The Euler-Lagrange equations corresponding to the minimization of (5) are:

$$0 = \frac{\partial f}{\partial x} - \frac{\partial}{\partial t}\frac{\partial f}{\partial \dot{x}} - \frac{\partial}{\partial \xi}\frac{\partial f}{\partial x'} + \frac{\partial^2}{\partial \xi^2}\frac{\partial f}{\partial x''} - \frac{\partial^2}{\partial \xi \partial t}\frac{\partial f}{\partial \dot{x}'} \tag{6}$$

$$0 = \frac{\partial f}{\partial y} - \frac{\partial}{\partial t}\frac{\partial f}{\partial \dot{y}} - \frac{\partial}{\partial \xi}\frac{\partial f}{\partial y'} + \frac{\partial^2}{\partial \xi^2}\frac{\partial f}{\partial y''} - \frac{\partial}{\partial \xi \partial y}\frac{\partial f}{\partial \dot{y}'} \tag{7}$$

Clearly however x and y are not indenendent, but are related by:

$$x'^2 + y'^2 = \ell^2 \tag{8}$$

Because the beam is inextensional. Performing the operations indicated in (6) and (7) on (5) and using (8) above (differentiated as often as necessary) the equation of motion of the beam becomes (eliminating x):

$$(\ddot{y} + \gamma^2 y^{iv})(1 - \lambda^2 y'^2)^2 = \lambda^2\gamma^2(2y'y''y''' + y''^3 + \lambda^2[y'^2 y''^3 - 2y'^3 y''y''']) \tag{9}$$

in which $\gamma^2 = EI/\rho\ell^4 A$.

INITIAL AND BOUNDARY CONDITIONS

The boundary conditions for the clamped edge at $\xi = 0$ are:

$$y(\xi = 0,t) = h(t) \qquad y'(\xi = 0,t) = 0 \tag{10}$$

(Clearly, it is somewhat tenuous to assume that the human torso is "built in" at the torso/hip joint. One might, instead, assume a hinged end instead of a clamped one at $\xi = 0$. The behavior of the torso/hip joint area is highly dependent upon the position of the individual during excitation, thus the choice of boundary condition may depend upon the circumstances of interest.)

The boundary conditions at the "free" end of the beam are expressed in terms of the properties of the lumped mass at that location, after a method developed by Karnopp [11]. For the mass:

$$\sum F_y = m\,\ddot{y}_{cg} \qquad \sum M_{cg} = J\,\ddot{\theta} \tag{11}$$

where the subscript cg refers to the mass center of the lumped mass. The y acceleration of the cg of the lumped mass can be expressed in terms of the beam's coordinates and θ as:

$$\begin{aligned} y_{cg} &= y(\xi = 1,t) + a \sin\theta \\ \dot{y}_{cg} &= \dot{y}(\xi = 1,t) + a \cos\theta\dot{\theta} \\ \ddot{y}_{cg} &= \ddot{y}(\xi = 1,t) + a(\cos\theta\ddot{\theta} - \sin\theta\dot{\theta}^2) \end{aligned} \tag{12}$$

using (12), expressions (11) become:

$$-M_{cg} = -M = ma\cos\theta\,\ddot{y} + (J + ma^2\cos^2\theta)\ddot{\theta} - ma^2\cos\theta\sin\theta\dot{\theta}^2 \tag{13}$$

$$-F_y = -V = m(\ddot{y} + a[\cos\theta\ddot{\theta} - \sin\theta\dot{\theta}^2]) \tag{14}$$

clearly θ can also be expressed in terms of the beam's coordinates:

$$\tan\theta = \left.\frac{dy}{dx}\right|_{\xi=1} = \left.\frac{\partial y}{\partial\xi}\Big/\frac{\partial x}{\partial\xi}\right|_{\xi=1} = \left.y'/x'\right|_{\xi=1} = \left.y'(\ell^2 - y'^2)^{1/2}\right|_{\xi=1} \tag{15}$$

For a beam in bending it is known that:

$$M(\xi,t) = -EIK(\xi,t) \tag{16}$$

$$V(\xi,t) = -\frac{1}{\ell}\frac{\partial M}{\partial\xi} = -\frac{EI}{\ell}\frac{\partial K(\xi,t)}{\partial\xi} \tag{17}$$

Evaluation of (16) and (17) at $\xi = 1$, together with substitution of the expressions (13) and (14) for V and M respectively yields the final form of the boundary conditions at the "free" end of the beam:

$$EIy'' = ma\ell\cos\theta(\beta\ddot{y} - \beta^{-1}a\sin\theta\,\dot{y}'^2) + \ell(J + ma^2\cos^2\theta)\left\{\frac{y'\dot{y}'^2}{\beta^2} + \ddot{y}'\right\} \tag{18}$$

$$\begin{aligned} EIy''' = {} & \frac{m\ell^4}{\beta}\left[\ddot{y} + a\cos\theta\left\{\frac{y'\dot{y}'^2}{\beta^3} + \frac{\ddot{y}'}{\beta}\right\} - a\sin\theta\,\frac{\dot{y}'^2}{\beta^2}\right] - \\ & - \frac{EI}{\beta^2}\left[(y'^2y''' + y''^2y') + \frac{y'^3y''^2}{\beta^2}\right] \text{ with } \beta^2 = \ell^2 - y'^2 \end{aligned} \tag{19}$$

where all boundary condition terms in (18) and (19) are evaluated at $\xi = 1$.

The specification of initial conditions is more straightforward: this consists of a specification on the initial displacements and velocities of the beam. Because the base or clamped end of the beam is subjected to a prescribed motion h(t) (corresponding, say, to the motion of the seat frame of a vehicle subjected to a rear-end collision), it is convenient to define a relative coordinate y_r such that:

$$y_r(\xi,t) = y(\xi,t) + h(t) \tag{20}$$

With this definition $y(\xi = 0,t) = 0$ and:

$$\begin{aligned} \frac{\partial^n y}{\partial \xi^n} &= \frac{\partial^n y_r}{\partial \xi^n} \\ \frac{\partial^n y}{\partial t^n} &= \frac{\partial^n y_r}{\partial t^n} + \frac{\partial^n h}{\partial t^n} \end{aligned} \tag{21}$$

The initial conditions of the system are then specified in terms of y_r as:

$$\begin{aligned} &y_r(\xi,0), \ 0 \leqslant \xi \leqslant 1 \\ &\dot{y}_r(\xi,0), \ 0 \leqslant \xi \leqslant 1 \end{aligned} \tag{22}$$

In terms of the relative coordinate y_r (9) can be rewritten as:

$$\ddot{y}_r + \ddot{h} = -\gamma^2 y_r^{iv} + \Omega(1 - \lambda^2 y_r'^2)^{-2} \tag{23}$$

in which $\Omega = \lambda^2\gamma^2(2y_r' y_r'' y_r''' + y_r''^3 + \lambda^2[y_r'^2 y_r''^3 - 2y_r'^3 y_r'' y_r'''])$

IDENTIFICATION OF PARAMETERS

The task of parameter identification is clearly made difficult in this case of dealing with living human subjects, together with their variability. Assuming that the use of a beam model for torso bending behavior is somewhat reasonable, and assuming that data on actual test subjects are available, the identification problem becomes that of determining values for the unknown parameters in the assumed model from measured data. In the present case, the parameters to be determined are γ and λ.

The elegant technique developed in the paper by Perdreauville and Goodson fails (at least if directly applied) in this case due to the triple product terms appearing in Ω above. It is possible that a clever rewriting of these terms, in some cases, might transform the equation into one upon which their technique will succeed [13].

In contrast to the analytical approach taken by [13], it is proposed to use a numerical technique developed by Tam [14]. The use of numerical techniques for parameter identification is not overly restrictive, particularly because the solution of the highly nonlinear differential equation (23) will in all likelyhood require further application of numerical techniques as well. The computational scheme employed takes the form of a discrete space, continuous time representation of the partial differential equation whose parameters are to be identified. Thus, a set of ordinary differential equations is formed, which are then quasilinearized and used to identify the unknown parameters of the distributed parameter system by minimizing the sum of the squares of the deviations between the observed data and the equation solution.

To begin, the spatial partial derivatives are replaced by their finite difference approximations. It is more convenient for purposes of the scheme to discrad the relative coordinate notation of (23) and return to the form of (9).

$$\begin{aligned} y'_i &= (y_{i+1} - y_{i-1})/2\Delta y \\ y''_i &= (y_{i+1} - 2y_i + y_{i-1})/(\Delta y)^2 \\ y'''_i &= (y_{i+2} - 3y_{i+1} + 3y_{i-1} + y_{i-2})/(\Delta y)^3 \\ y^{iv}_i &= (y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2})/(\Delta y)^4 \end{aligned} \tag{24}$$

This procedure gives one ordinary differential equation for each of the n interior grid points y_i. The resulting set of differential equations is of the form:

$$\frac{d^2 y(t)_i}{dt^2} = f_i(y_{i+2}, \ldots, y_{i-2}, \Delta y, \gamma, \lambda) \qquad i = 1,2, \ldots, n \tag{25}$$

Quasilinearization is then used to obtain an estimate for the solution of (25) at the next iteration for γ and λ (indicated by the superscript k+1). This is done by expanding about the values of y_i and $U_1, U_2, \ldots, U_m$, where U represents the vector of undetermined parameters.

$$\frac{d^2 y_i^{k+1}}{dt^2} = f_i^k + \sum_{j=1}^{n}(y_j^{k+1} - y_j^k)\left(\frac{\partial f_i}{\partial y_j}\right)^k + \sum_{j=1}^{m}(U_j^{k+1} - U_j^k)\left(\frac{\partial f_i}{\partial U_j}\right)^k \qquad i = 1,2, \ldots, n \tag{26}$$

The general solution of (26) can be expressed as a sum of the particular solution plus a linear combination of the complimentary solutions, such that:

$$y_i^{k+1} = P_i(t) + \sum_{j=1}^{m} U_j^{k+1} H_{ij}(t) \qquad i = 1,2, \ldots, n \tag{27}$$

Substituting (27) into (26) and equating coefficients gives:

$$\frac{d^2 P_i}{dt^2} = f_i^k + \sum_{j=1}^{n}(P_j - y_j^k)\left(\frac{\partial f_i}{\partial y_j}\right)^k - \sum_{j=1}^{m} U_j^k\left(\frac{\partial f_i}{\partial U_j}\right)^k \qquad i = 1,2, \ldots, n \tag{28}$$

and

$$\frac{d^2 H_{ij}}{dt^2} = \sum_{\ell=1}^{n} H_{\ell j}\left(\frac{\partial f_i}{\partial y_\ell}\right)^k + \left(\frac{\partial f_i}{\partial U_j}\right)^k \qquad i = 1,2, \ldots, n,\ j = 1,2, \ldots, m \tag{29}$$

together with the initial conditions $P_i(0)$, $P_i(0)$, $i = 1,2, \ldots, n$ and $H_{ij}(0) = 0$, $i = 1,2, \ldots, n$, $j = 1,2, \ldots, m$, as well as $\dot{H}_{ij}(0) = 0$ for the same subscripts.

The objective is to minimize the sum of the squares of the deviations of observed data $D_i(t)$ and the computed values $y_i(t)$ at some location i. (This can, of course, be done for several locations.) Mathematically:

$$S = \sum_{t=1}^{T}(y_i - D_i)^2 \tag{30}$$

and (30) is to be minimized with respect to all of the parameters to be identified. For a minimum:

$$\frac{\partial S}{\partial U_1} = \frac{\partial S}{\partial U_2} = \cdot \cdot \cdot = \frac{\partial S}{\partial U_m} = 0 \tag{31}$$

But from (27) S can be written:

$$S = \sum_{t=1}^{T}([P_i + U_1^{k+1}H_{i1} + \ldots + U_m^{k+1}H_{im}] - D_i)^2 \tag{32}$$

By taking the first derivative of (32) with respect to U_1 through U_m respectively and setting each of these to zero, a set of m algebraic equations for the (k+1)st values of the unknown parameters U_m is formed. These have the form:

$$\sum_{t=1}^{T}([P_i + U_1^{k+1}H_{i1} + \ldots + U_m^{k+1}H_{im}] - D_i)H_{ij} = 0 \qquad j = 1,2, \ldots ,m \tag{33}$$

or in matrix shorthand:

$$A\underline{U} = \underline{B} \tag{34}$$

in which A is an mxm symmetric matrix and $\underline{U}$ is the vector of parameters to be determined. $\underline{B}$ is a constant vector formed of terms which are products of D_i, H_{ij} and P_i.

The cycle is started by making initial estimates for the values of the unknown parameters $\underline{U}$. The particular and homogeneous ordinary differential equations of (28) and (29) are then integrated numerically and the resulting values substituted into the matrix equation (34), which yields, upon solution the (k+1)st values of the parameters, i.e., a new estimate for their values. This procedure is repeated until the sum of the squares of deviations S reaches an acceptable minimum value.

Tam [14] has demonstrated the efficiency of this technique in his work. Further, he show d an extension of this technique applicable to determination of unknown boundary conditions. Suffice to say that this extension may be of considerable value in the problem at hand due to the uncertainty of the boundary conditions which exist at the torso/hip joint area.

SUMMARY

The differential equation of motion and attendant boundary conditions for a distributed parameter model of human sagital plane torso bending behavior has been developed and a parameter identification technique applicable to the determination of its parameters shown. Comparison of the above mathematical representation with actual data will be the subject of a second paper.

REFERENCES

1. McHenry, R. R., "Computer Simulation of the Automobile Crash Victim in a Frontal Collision: A Validation Study," Cornell Aeronautical Laboratory, Report No. YB-2126-V-1R, Buffalo, New York, 1966.

2. Patrick, L. M. and Mertz, H. J., Jr., "Impact Dynamics of Unrestrained, Lap Belted and Lap and Diagonal Chest Belted Vehicle Occupants," Proceedings of the 10th Stapp Car Crash Conference, 1966.

3. Naab, K. N., "A Computer Simulation of the Automobile Crash Victim," Cornell Aeronautical Laboratory, Report No. VJ-1823-R18, Buffalo, N.Y., 1966.

4. Liu, Y. K., "The Human Body Under Time-Dependent Boundary Conditions," Progress Report for Period September 1, 1966 - March 15, 1968, Department of Engineering Mechanics, University of Michigan, Ann Arbor, Michigan, 1968.

5. Jacobsen, L. S. and Ayre, R. S., Engineering Vibrations, McGraw-Hill Book Company, New York, New York, 1958, pp. 482-483.

6. Timoshenko, S., "On the Transverse Vibrations of Bars of Uniform Cross Section," Philosophical Magazine, Serial VI, Volume 43, 1922.

7. Anderson, R. A., "Flexural Vibrations in Uniform Beams According to the Timsohenko Theory," Transactions of the ASME, Journal of Applied Mechanics, vol. 20, 1953.

8. Wagner, H., "Large Amplitude Free Vibrations of a Beam, Transactions of the ASME, Journal of Applied Mechanics, vol. 32, 1965.

9. Weinstock, R., Calculus of Variations, McGraw-Hill Book Company, New York, New York, 1952, pp. 32-36.

10. Timoshenko, S., Vibration Problems in Engineering, 3rd Edition, D. Van Nostrand Company, Inc., Princeton, New Jersey, 1955, p. 334.

11. Karnopp, D., Random Vibrations, Vol. II (S. Crandall, Ed.), MIT Press, Cambridge, Massachusetts, 1963, chapter I.

12. Ho, B. L. and Kalman, R. E., "Effective Construction of Linear Space Variable Models from Input-Output Functions," Regelungstechnik, vol. 14, 1966.

13. Perdreauville, F. J. and Goodson, R. E., "Identification of Systems Described by Partial Differential Equations," Transactions of the ASME, Journal of Basic Engineering, vol. 88, 1966.

14. Tam, W. C., "Identification of Parameters in Distributed Parameter Systems Using Quasilinearization and Finite Differences," Proceedings of the 1972 Summer Computer Simulation Conference, San Diego, California, 1972.

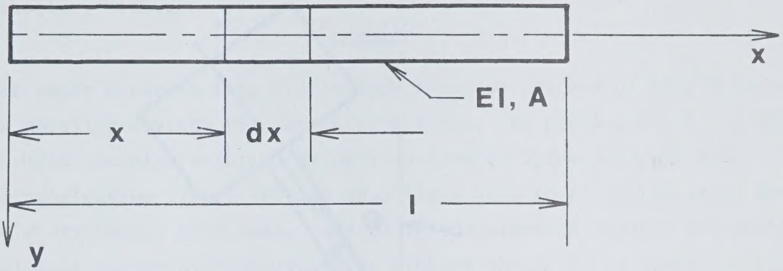


Figure 1

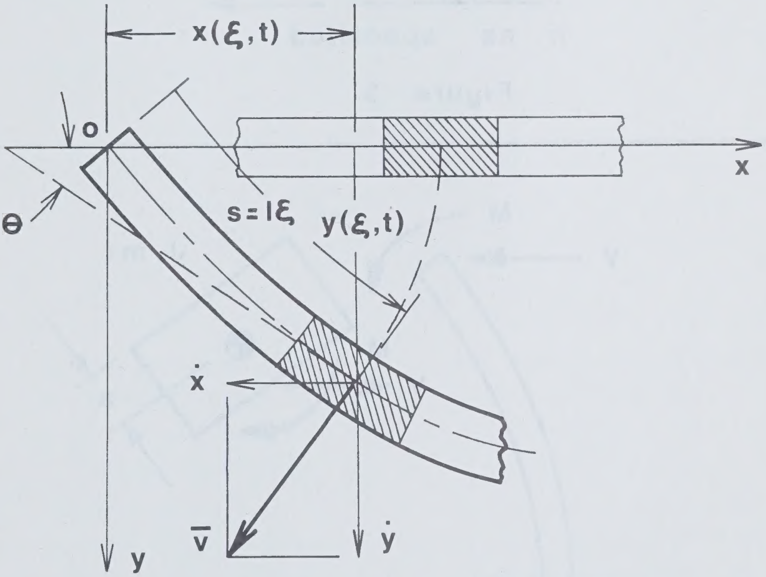
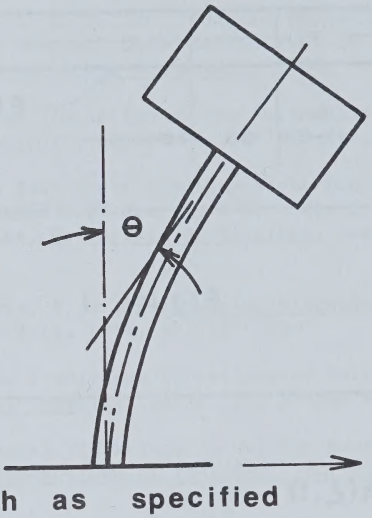

x(ξ,t)
o
x
θ
s = lξ
y(ξ,t)
ẋ
v̄
ẏ
y

Figure 2


θ
h as specified

Fiyure 3

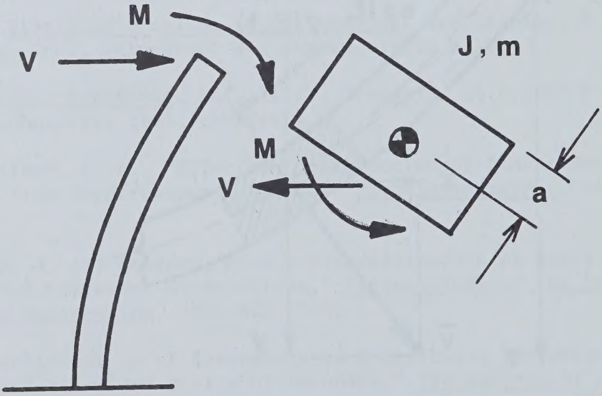


Figure 4

HYBRID IMPLEMENTATION OF PARTIAL DIFFERENTIAL EQUATIONS FOR PARAMETER IDENTIFICATION AND MODEL MATCHING

MICHAEL J. WOZNY
School of Engineering
Oakland University
Rochester, Michigan

SUMMARY

This paper presents some preliminary ideas on the use of hybrid computers for modeling distributed parameter systems. In particular, it is shown that a hybrid computer solution using a continuous space-discrete time (CSDT) approximation offers certain advantages when modeling the axial temperature distribution of a lake. The CSDT approximation reduces the problem to a nonlinear two-point boundary value problem which can be solved using a decomposition of operators approach with a predictor-corrector algorithm. Detailed results will be published later.

A TECHNIQUE FOR PARAMETER IDENTIFICATION IN A CLASS OF HYPERBOLIC DISTRIBUTED PROCESSES

R. E. KLEIN
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

ABSTRACT

Hyperbolic or wave like distributed parameter processes occur in a number of physical settings. Examples are torsional bending in sea going ships, torsional bending during laying of deep sea pipelines to offshore drilling rigs, and foundation piles which are being driven into place during the construction of tall buildings. An underlying problem is the determination of the system parameters from on-line measurement data. An identification technique is discussed in this paper which is based on orthogonality relationships of these hyperbolic processes in the time variable. The paper discusses and interprets the parameter identification process, its measurement requirements, and examples of both identifiable and non-identifiable processes.

INTRODUCTION

A number of physical hyperbolic problems arise in engineering where it is desirable to determine the parameters of the distributed process. This section of the paper will be devoted to a specific discussion of several of these processes. Following that certain observability results for one dimensional hyperbolic partial differential equations will be examined. This paper then concludes with an examination of, and presentation of, a class of parameter identification techniques for second order linear hyperbolic processes.

Torsional Bending in Seagoing Ships.

Ships at sea are subject to stresses and inputs from a variety of sources. As ship lengths increase, the longitudinal torsion of the hull becomes an increasingly important factor affecting ship integrity. (See Abrahamsen [1].) Klein [2], and Klein and Hughes [3] present discussions of the hyperbolic observability and distributed parameter control problems associated with large seagoing ships. In the previous works [2], [3], it was assumed that the hyperbolic wave speed was constant and uniform throughout the ship length thereby resulting in a normalized uniform wave equation. A number of physical settings occur in which the propagation speed varies with spatial distance and is also unknown. In these cases, a parameter in a hyperbolic wave equation model is unknown. Physically, the need to determine this parameter would result upon initial loading of a ship, or more severely, due to an unforeseen load shift at sea, or a partial structural failure. Basically, the mathematical problem amounts to the determination of the parameter c(x) in the mathematical model

$$u_{tt}(x,t) = c^2(x)\, u_{xx}(x,t) \tag{1}$$

for $0 < x < L$, $0 \leq t \leq T$, and

$$u_x(0,t) = u_x(L,t) = 0, \quad 0 < t < T \tag{2}$$

where $c(x) = [I(x)/GJ(x)]^{1/2}$, GJ(x) is the torsional stiffness, and I(x) is the torsional mass moment of inertia. A torsional excitation term f(x,t) has been omitted in Eq. (1). It is often small in magnitude and significant torsional oscillations usually occur only after a succession of periodic inputs (such as wave swells). Thus, the effect for $t < 0$ of the forcing function can be considered as an unknown set of initial conditions in Eq. (1). Furthermore, the absolute roll displacement $\hat{u}(x,t)$ at any point and time is the sum of the rigid body roll $r_r(t)$ and the torsional deflection u(x,t); *i.e.*, $\hat{u}(x,t) = u_r(t) + u(x,t)$. Now, $u_r(t)$ is governed by a second order differential equation of the form

$$\ddot{u}_r(t) + 2\hat{\zeta}\hat{\omega}_n\, \dot{u}_r(t) + \hat{\omega}_n^2\, u_r(t) = f_o(t) \tag{3}$$

where $\hat{\zeta}$ is the effective roll damping coefficient and $\hat{\omega}_n$ is the natural frequency of roll oscillation. Since the techniques and parameters for rigid body roll control are well known, for parameter identification it suffices to consider Eq. (1).

Laying of Deep Sea Pipelines

As off shore drilling rigs are located in deeper water, such as in the Gulf of Mexico and in the North Sea, the limits to which deep sea pipelines can be used are presently determined by the ability to lay the pipelines. Customarily, the pipe is laid in a continuous length, from a ship, or ships, where the pipeline is welded into a continuous section and lowered as the ship moves forward. As a result, the pipeline hangs in a suspended shape until it settles to rest on the sea bed. Unfortunately, the length of suspended pipe, which is usually sinking slowly, is subjected to dynamics from a variety of sources, particularly from Von Karman vortices as the pipe sinks, ocean currents, and movements of the pipelaying ship due to wind and sea swells. The application of active controls for pipe laying has been investigated, however, the parameter identification problem is paramount. In particular, the actual length of the suspended pipeline is unpredictable, and adequate techniques do not exist for the determination of the parameter, suspended length. Consequently, the following hyperbolic wave equation model is applicable where the length L(t) is the unknown parameter,

$$u_{tt}(x,t) = c^2 u_{xx}(x,t) + f(x,t) \tag{4}$$

$$0 < x < L(t), \qquad 0 \leq t \leq T,$$

and where

$$u(0,t) = \phi_1(t),\ u(L,t) = 0,\quad 0 \leq t \leq T \tag{5}$$

Active controls can be envisioned, which might consist of, for example, manipulated fins at various intervals on the pipe. In this case, the manipulation of fins would be sufficient to exert appropriate moments, and thereby control, on the pipe as it sinks in the water. However, such a scheme would require a distributed state estimation process, which in turn, would require simultaneous estimation of the parameter L(t).

Pile Driving

The weight requirements of many modern structures require that foundation piles be driven into the soil beneath the structure. In certain locales, piles are driven until bedrock is reached, and a firm support is guaranteed. However, many soil structures do not have an accessible underlying bedrock and it is essential to drive piles in a manner such that satisfactory bearing load is achieved. At present, it is standard practice to take a number of core drillings to determine the subsoil structure. Unfortunately, the adequate prediction of bearing load of a pile from core samples is not entirely satisfactory. Cases occur when piles inadvertently "punch" a load bearing soil layer with the result that bearing load diminishes drastically. Also, a pile will alter, through a process not fully understood, its bearing load ability for a period of time following the pile's installation. In cases suspected of this, it is then necessary to later test the pile statically or with a single dynamic tap.

The pile driving process could be aided and improved if adequate techniques for on-line parameter identification could be achieved. Specifically, it would be desirable to determine the parameter b(x) in the wave equation model

$$u_{tt}(x,t) = c^2 u_{xx}(x,t) + b(x)\, u_x(x,t) \tag{6}$$

for

$$0 \leq x \leq L, \qquad 0 \leq t \leq T \tag{7}$$

Since b(x) is directly related to the shear load bearing ability, the number of physical measurements on the pile would normally be restricted and would normally be located on the upper exposed portion of the pile. The pile driving parameter identification problem is representative of a difficult class of distributed parameter identification problems.

PARAMETER IDENTIFICATION

Observability and State Estimation

Observability and state estimation results for one-dimensional wave equation processes have been given in Klein and Hughes [3], and in Goodson and Klein [4]. The fundamental facts are that observability and state estimation are well resolved for linear one-dimensional processes with constant and known parameters. Specifically, if c(x) in Eq. (1) is known and constant, considering the ship torsion problem, then the following results hold.

Theorem 1. For $0 \leq x \leq 1$, $0 \leq t \leq T$, and $u(x,t) \in C^2$, then

$$|u(x,t)| \leq 4c_1(N,x_m)\left\{ \sup_{0 \leq t \leq T} |u_x(x_m,t)| \right\} + [2c_o(N)/\pi^2][M_4 + M_5] \tag{8}$$

where $c_o(N)$ and $c_1(N,x_m)$ are bounded computable constants given in [3], and where

$$u_x(x_m,t) = \int_0^1 \delta(x - x_m)\, u_x(x,t)\, dx$$

$$0 \leq T_o \leq t \leq T_m \leq T \tag{9}$$

The proof is given in [3], however, it is based on orthogonality properties of the wave equation solution from which one obtains

$$\begin{Bmatrix} a_n \\ b_n \end{Bmatrix} = \int_0^1 \begin{Bmatrix} (1/n\pi)\, u_t(x,0) \\ u(x,0) \end{Bmatrix} \cos(n\pi x)\, dx \tag{10}$$

where a_n and b_n, $n = 1,2,3,\cdots$ are the Fourier expansion coefficients in the solution

$$u(x,t) = \sum_{n=1}^{\infty} [a_n \sin (n\pi t) + b_n \cos (n\pi t)] \cos (n\pi x) \qquad (11)$$

Also, based on Theorem 1, it is possible to state the following corollary.

Corollary 1. Observation of the data $g(t)$ for $T_o \leq t \leq T_m$, where $T_m = T_o + 2$, is necessary and sufficient for N-mode observability of $u(x,t)$ provided

$$\sin (n\pi x_m) \neq 0, \quad n = 1,\cdots,N. \qquad (12)$$

In addition, similar but extended results hold for the case of Eq. (1) with a damping term added. These details are given in [3].

Parameter Estimation

The above observability results permit a number of parameter estimation results to be stated. Specifically, the orthogonality relationships, such as Eq. (10) permit direct determination of the distributed parameter c in Eq. (1). This fact can be formalized with the following theorem.

Theorem 2. A single measurement $g(t)$ in the process

$$u_{tt}(x,t) = c^2 u_{xx}(x,t), \quad 0 < x < L, \quad 0 \leq t \leq T \qquad (13)$$

where

$$u(0,t) = u(L,t) = 0, \quad 0 \leq t \leq T \qquad (14)$$

and where the measurement $g(t)$ at a measurement location x_m is given by

$$g(t) = \int_0^L \delta(x - x_m) u_x(x,t) dx, \; 0 \leq t \leq T_m \qquad (15)$$

is sufficient for the unique determination of c, where $c > 0$, given

(i) The bounded initial state $u(x,0)$ and $u_t(x,0)$ of the process,

(ii) C^*, an *a priori* lower bound on c such that $T_m \geq 2L/C^*$, and

(iii) $\sin n\pi x_m \neq 0, \quad n = 1,2,3,\cdots$.

Furthermore, several equally important theorems can be stated:

Theorem 3. Given a process and only the measurement satisfying Eqs. (13), (14), and (15) above, it is not possible to uniquely determine c in the absence of the information (i), *i.e.*, knowledge of the initial state of the process.

Theorem 4. A set of two or more measurements $g_1(t), g_2(t), \cdots, g_N(t)$ given by

$$g_j(t) = \int_0^L \delta(x - x_{m_j})\, u_x(x,t)\, dx \qquad (16)$$

for $j = 1, \cdots, N$ is insufficient for the unique determination of c in Eq. (13) in the absence of (i).

CLOSURE

Theorem 3 can be proved easily by demonstrating that two different sets of c, $u(x,0)$ and $u_t(x,0)$ fail to provide sufficient conditions to determine c. Theorem 4 becomes obvious upon insertion of Eq. (11) into the measurement equation which demonstrates that the additional measurements provide only redundant information to the original measurement $g(t)$.

The problems of state estimation and parameter identification in hyperbolic distributed processes continue to challenge controls and systems engineers. The brief results indicated in this paper provide both positive and negative results, since certain conditions for identifiable, and likewise, non-identifiable hyperbolic processes and measurements are outlined. From these results, it appears likely that the measurement requirements, in the absence of initial condition information, will be exceedingly stringent. In fact, corresponding results for parabolic equations, *e.g.*, the heat equation, suggest that mathematical non-uniqueness will be more often the case than uniqueness. A collection of references related to similar problems for the heat equation are given in [5].

REFERENCES

1. Abrahamsen, E., "Structural Design Analysis of Large Ships," Trans. of The Society of Naval Architects and Marine Engineers, 1955.

2. Klein, R. E., "Non Well Posed Problems in the Sense of Hadamard and Their Relationship to Distributed Parameter State Reconstructor Problems," Proceedings of The IFAC Symposium on the Control of Distributed Parameter Systems, Banff, Canada, June 1971, Paper No. 7-2.

3. Klein, R. E. and Hughes, R. O., "The Distributed Parameter Control of Torsional Bending in Seagoing Ships," Paper No. 8-B4, Preprint Volume of the 1971 J.A.C.C., St. Louis, Mo., August 1971, pp. 867-875.

4. Goodson, R. E. and Klein, R. E., "A Definition and Some Results for Distributed System Observability," IEEE Trans. on Automatic Control, April 1970, pp. 165-174.

5. Cannon, J. R. and Klein, R. E., "Optimal Selection of Measurement Locations in a Conductor for Approximate Determination of Temperature Distributions," Trans. of ASME, J. of Dynamic Systems, Measurement and Control, September 1971, pp. 193-199.

IMPLANT ELECTRONICS FOR MEDICAL APPLICATIONS

W. H. Ko
Case Western Reserve University
Cleveland, Ohio

Abstract

Implant electronics can be utilized in medical research, clinical monitoring, and many other applications. Selected examples will be discussed to illustrate possible uses of implantable telemetry and stimulation devices. Major problems in design and implant devices will be outlined with examples in single channel and multiple channel micropower telemetry systems, active transducers for implant instruments, as well as passive and active implant stimulators.

INTEGRATED CIRCUITS FOR BIO-MEDICAL APPLICATIONS

JOSEF BERGER*
Stanford University
Stanford, California 94305

ABSTRACT

Several new integrated circuits were developed at Stanford Integrated Circuits Laboratory for a unique biomechanical application: the reading aid for the blind. The circuits perform image sensing, information processing and storage, and driving of piezoelectric transducers. The technology of integrated circuits played a vital role in this application.

PRINCIPLE OF THE READING AID

The reading aid has been extensively described in the literature [1], and we will limit ourselves here to a very brief description. It is a non-translating reading device, which means that it presents to a blind person true letter shapes in a form of tactile images. The right hand guides a little solid-state camera along the line of regular printed material (see Fig. 1). The output tactile image is formed by an array of piezoelectric transducers (one element is shown in Fig. 2), and it is displayed on the tip of the left index finger. The camera contains a solid-state image sensor, illuminators and zoom optics (see Fig. 3). The information is processed in the electronic system, the block diagram of which is shown in Fig. 4. The complete reading aid weighs about 4 pounds. It is a portable, battery-operated instrument. At the present time it is being used by more than 100 persons all over the world, with reading speeds as high as 90 words per minute. The operation can be described as follows (see Fig. 4).

The 6 x 24 phototransistor array and the corresponding array of the output switching transistors are driven synchronously from a shift register. The image information is taken out in six parallel channels and fed into comparators. The binary comparator output (dark or light) directly controls the switching transistors which drive piezoelectric transducers. This schematic shows the basic parts of the reading aid. Two of them--the image sensor and the transducer drive circuit--were developed in the monolithic integrated circuit form. The rest of the electronic system uses integrated circuits readily available on the market.

IMAGE SENSOR

The bipolar image sensor [2] is an array of 6 x 24 phototransistors (Fig. 5). Its electrical schematic is shown in Fig. 6. The collectors are connected to the columns (A ÷ F) and the emitters to the rows (1 ÷ 24). The transistor bases are floating; pn junction isolation between columns is used. The phototransistors operate in charge storage mode (Fig. 7). Photoresponses of one column of the array at three different light levels are shown in Fig. 8. The uniformity is better than ±5% in the dynamic range of 20 db. The clock frequency is 1.5 kHz in this case. The responsivity of the sensor is 20 nA/μw at tungsten illumination 2854° K. The spectral response was optimized to the wavelength of 5500 Å. It is a relatively large integrated circuit (2.4 x 3.8 mm), and great care was exercised to insure high production yields. This photosensor is used in the present model of the reading aid, which is manufactured by Telesensory Systems, Inc., in Palo Alto.

TRANSDUCER DRIVE CIRCUIT

This circuit contains a twelve-stage dynamic shift register, an array of MOS memory capacitors and twelve drive transistors [4]. The circuit is depicted in Fig. 9. This is an MOS circuit--in this particular case, Al gate type. The output transistor switches about 50V needed for driving of the piezoelectric transducers. The MOS memory capacitors have a total capacity of approximately 120 pF. Fig. 10 shows one stage of the circuit (without a shift register). To illustrate some problems which had to be solved, let us mention two phenomena which endanger the circuit operation [5]:

---

*Now with Hewlett-Packard Laboratories, Palo Alto, California

(1) Parasitic lateral PNP transistor exists between switching transistor T_s and the information transfer transistor T_i. Although the current gain of this transistor was of the order of 0.001, this value was high enough to discharge the memory capacitor C_M and cause the circuit to malfunction. The problem was solved by using special circuit geometry, enlarging the injection-collection distance and collecting the excessive charge (see Fig. 9).

(2) The gate of the output switching transistor T_s is floating during most of the operation, and its potential is given by memory capacitor charge (see Fig. 10). Due to the existence of parasitic capacity C_p the gate voltage V_G can vary with the T_s drain voltage. This is the second parasitic phenomenon. It can be controlled, for example, by using a circuit from Fig. 11. Here the two output transistors are used in series. This decreases the capacitive feedback, and it also increases the switching voltage. The circuit is shown in Fig. 12.

Fig. 13 shows the Si gate transducer drive circuit schematic. This circuit operates using a unique variable feedback scheme which permits a part of the drain voltage to drive the gate, thus increasing significantly the saturation current and at the same time saving the circuit area [6]. The finished circuit is shown in Fig. 14. Combining this feedback and the high voltage Si gate output transistors (see Fig. 15), we have designed a new transducer drive circuit operating at 100V, having a significantly smaller area (~75%) and operating about 10 times faster. This circuit will be used in the future MOS version of the reading aid.

CONCLUSION

The purpose of this paper was to show the role of integrated circuits in one unique bio-medical application and to show what sort of electronic problems had to be solved in order to make a useful reading device.

REFERENCES

[1] Linvill, J.G., and J.C. Bliss, "A Direct Translation Reading Aid for the Blind," Proc. IEEE, Vol. 54, No. 1, p. 40, January 1966.

[2] Salsbury, P.J., and J.D. Meindl, "A Monolithic Image Sensor for a Reading Aid for the Blind," Proc. IEEE, Vol. 58, No. 9, p. 409, Sept. 1970.

[3] Meindl, J.D., et al., "Integrated Electronics for a Reading Aid for the Blind," Digest of 1969 ISSCC, p. 50, February 1969.

[4] Plummer, J.D., and J.D. Meindl, "MOS Electronics for a Reading Aid for the Blind," Digest of 1970 ISSCC, p. 168, February 1970.

[5] Plummer, J.D., J. Berger and J.D. Meindl, "High Voltage Monolithic MOS Drive Arrays," Digest of 1971 IEDM, Washington, D.C., October 1971.

[6] Berger, J., "Improved Transducer Drive Circuit," to be published in J. Solid State Circuits.

[7] Berger, J., "Circuits for a Reading Aid for the Blind," IEEE Int. Symp. on Circuit Theory, April 1972.

This work was supported by Office of Education Grant 0-8-071112-2995 of the Department of Health, Education, and Welfare. However, the opinions expressed herein do not necessarily reflect the position or policy of the U.S. Office of Education, and no official endorsement by that office should be inferred.

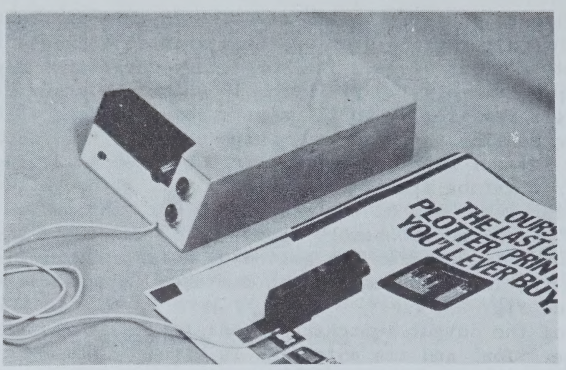

OURS
THE LAST
PLOTTER PRIN
YOU'LL EVER BUY.

Fig. 1. The Reading Aid for the Blind.

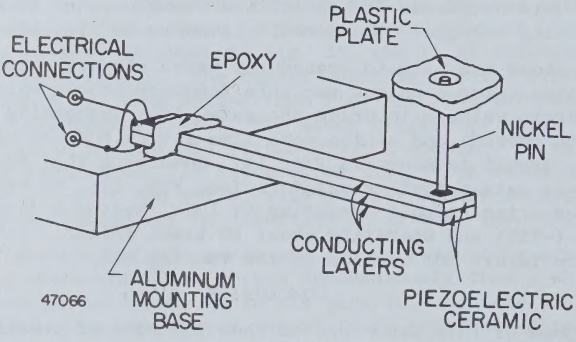

PLASTIC PLATE
ELECTRICAL CONNECTIONS
EPOXY
NICKEL PIN
CONDUCTING LAYERS
ALUMINUM MOUNTING BASE
47066
PIEZOELECTRIC CERAMIC

Fig. 2. One Element of the Tactile Display.

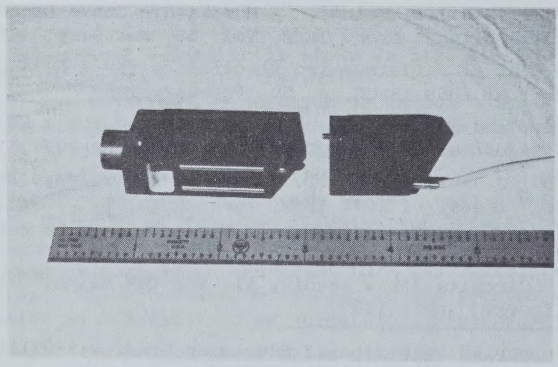


Fig. 3. Optical and Sensor Part of the Camera.

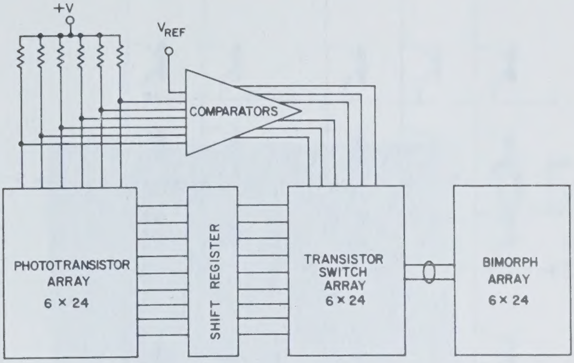

+V
V_{REF}
COMPARATORS
PHOTOTRANSISTOR ARRAY 6 × 24
SHIFT REGISTER
TRANSISTOR SWITCH ARRAY 6 × 24
BIMORPH ARRAY 6 × 24

Fig. 4. Block Diagram of the Reading Aid.

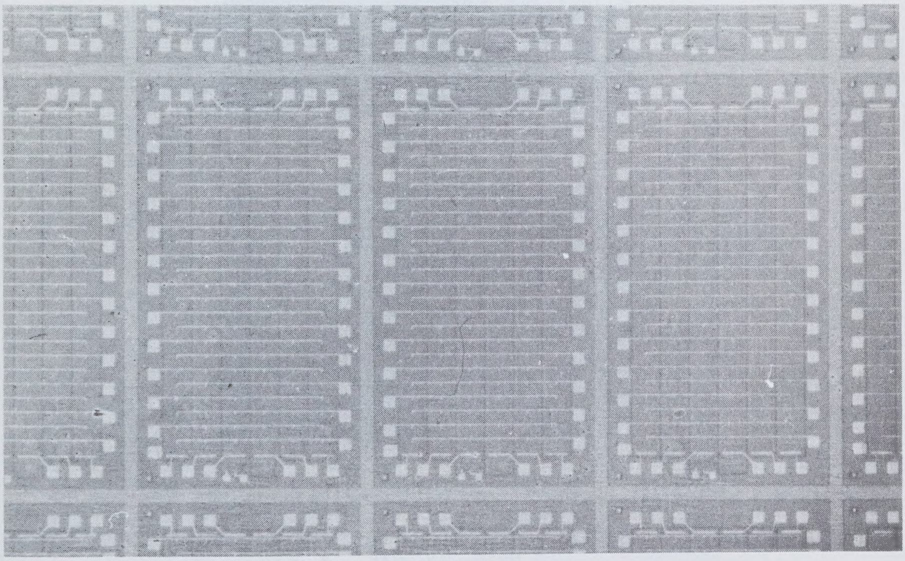


Fig. 5. The Bipolar Image Sensor.

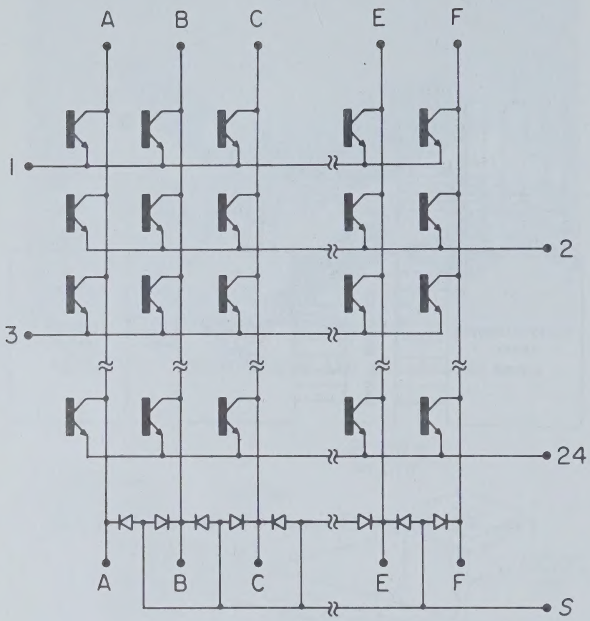

A
B
C
E
F
1
2
3
24
A
B
C
E
F
S

Fig. 6. Electrical Schematic of the Bipolar Image Sensor.

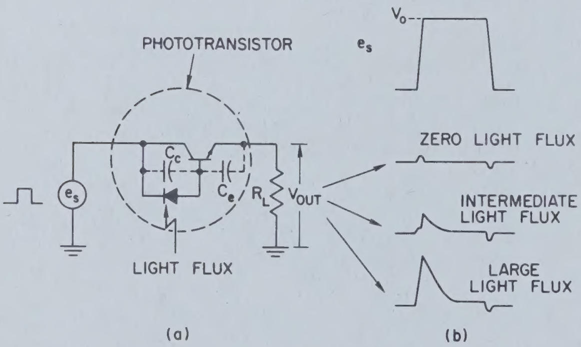

PHOTOTRANSISTOR
C_c
C_e
e_s
R_L
V_{OUT}
LIGHT FLUX
V_o
e_s
ZERO LIGHT FLUX
INTERMEDIATE LIGHT FLUX
LARGE LIGHT FLUX
(a)
(b)

Fig. 7. Charge Storage Mode Operation of a Phototransistor.

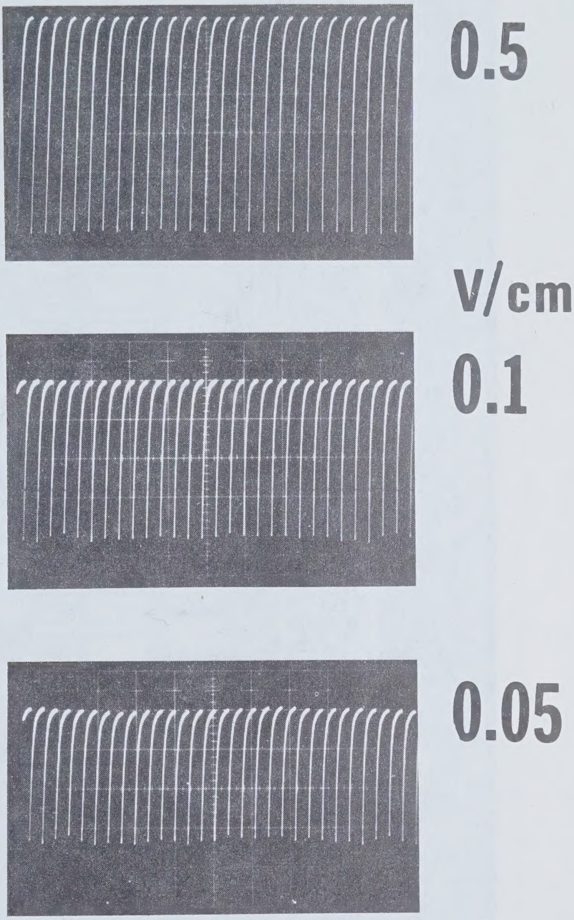


Fig. 8. Bipolar Image Sensor Response at Three Different Light Levels.

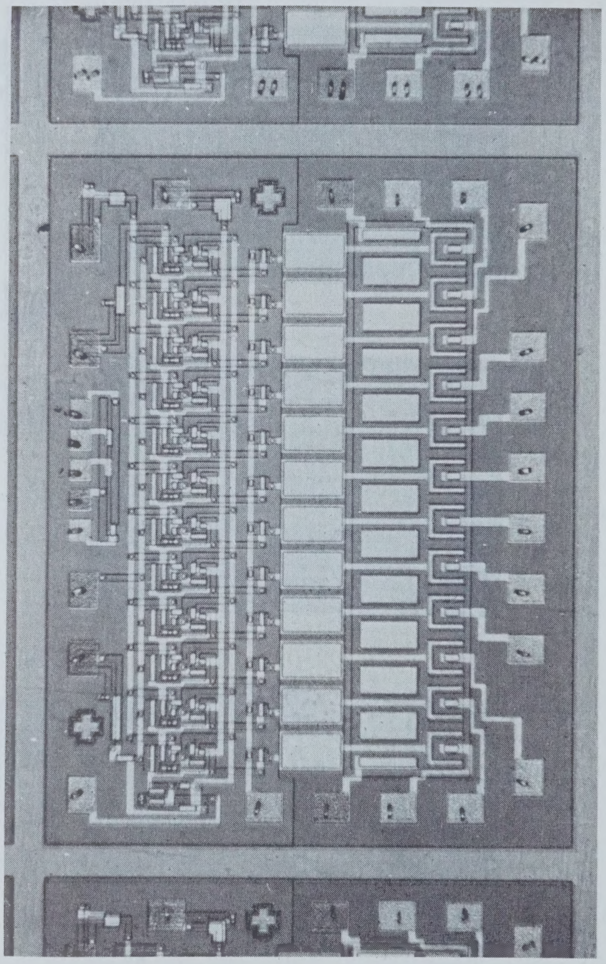


Fig. 9. Transducer Drive Circuit.

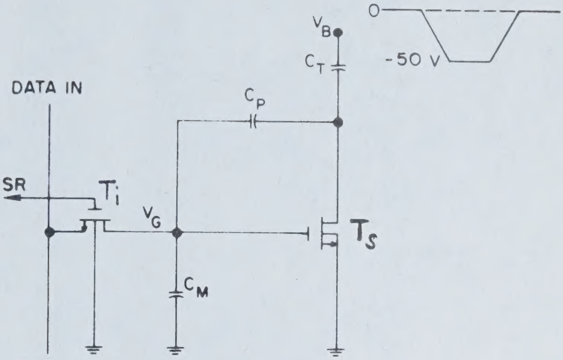

0
-50 V
V_B
C_T
C_P
DATA IN
SR
T_i
V_G
T_S
C_M

Fig. 10. One Stage of the Transducer Drive Circuit.

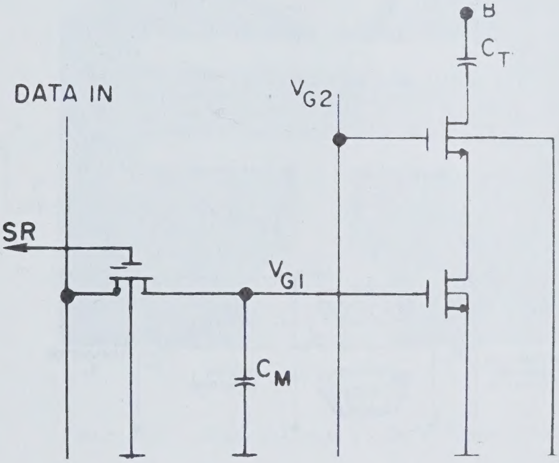

B
C_T
DATA IN
V_{G2}
SR
V_{G1}
C_M

Fig. 11. Two-Transistor Output Circuit

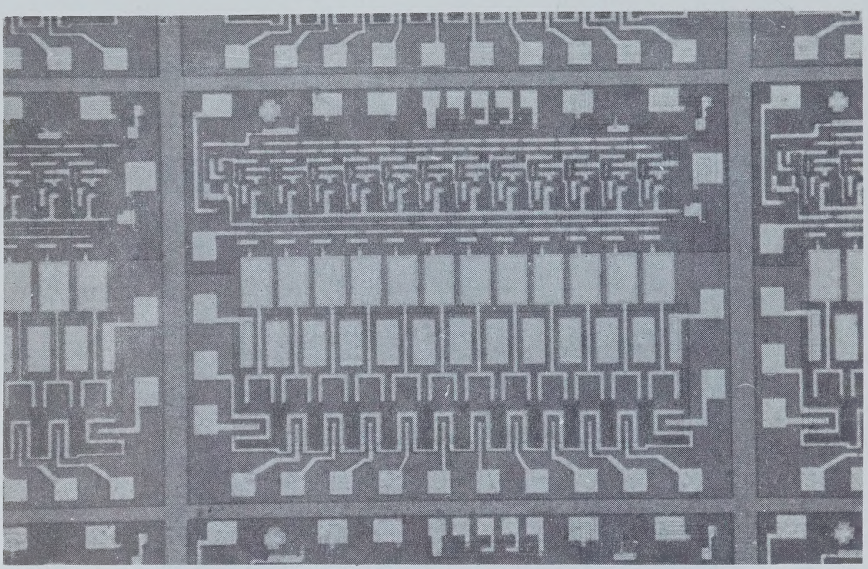


Fig. 12. Improved Transducer Drive Circuit.

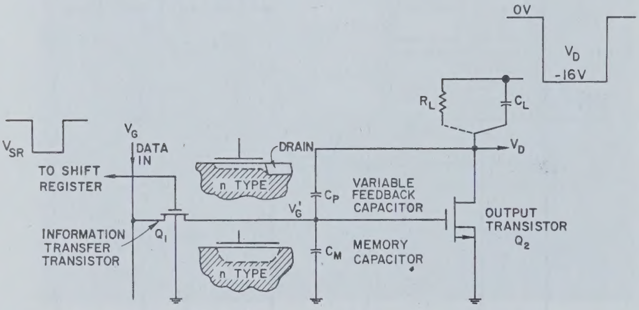

OV
V_D
-16V
R_L
C_L
V_D
V_SR
V_G
DATA IN
TO SHIFT REGISTER
DRAIN
n TYPE
C_P
VARIABLE FEEDBACK CAPACITOR
V'_G
INFORMATION TRANSFER TRANSISTOR
Q_1
n TYPE
C_M
MEMORY CAPACITOR
OUTPUT TRANSISTOR Q_2

Fig. 13. Variable Feedback Transducer Drive Circuit.

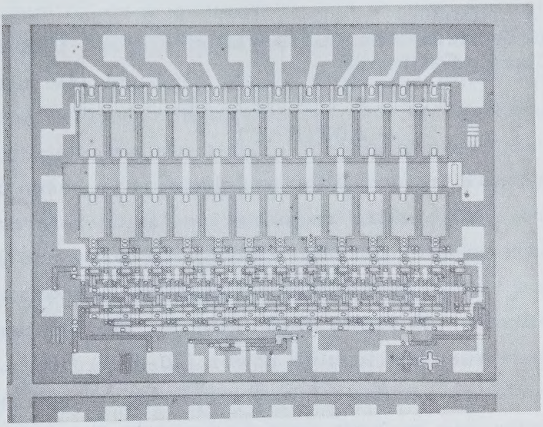


Fig. 14. Si Gate Transducer Drive Circuit.

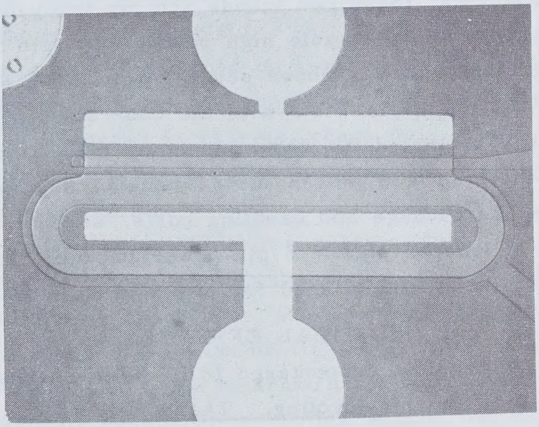


Fig. 15. High Voltage Si Gate Transistor.

NEW DEVELOPMENTS IN MONOLITHIC VOLTAGE REGULATORS

CARL T. NELSON
National Semiconductor Corporation
Santa Clara, California

ABSTRACT

Improvements in monolithic regulators have been rapid in the past few years. The entire regulator including reference, high current pass transistor, current limiting networks and loop stabilizing capacitors is now formed on a single monolithic chip. Advances in circuit and layout techniques as well as packaging improvements have resulted in high power (20-30W) single chip regulators which are virtually blow-out proof. A new synthesized voltage reference can be built as a two terminal device with 1.2V output for operation from low voltage supplies.

TRENDS

The most recent trends in monolithic regulators have been toward higher power and reduced external component count. Because of package problems, these two trends are not independent, but reflect the lack of an acceptable high power multi-pin regulator package. For power levels above several watts, the standard package today is the 2-pin TO-3 with the case serving as a third lead. This has prompted the development of monolithic regulators complete on-the-chip with only 3 leads to the outside world. In an attempt to reduce package and handling costs, several manufacturers have recently developed plastic 3 pin power packages for these regulators, similar to those used for power transistors.

THERMAL EFFECTS

The worst problem encountered in IC power packages is fatigue failure created by thermal cycling. These failures can occur at four places in the package; at the die to interface junction, at the interface to package junction, at the bonding wire to chip junction and at the package to terminal seal. Thermal fatigue occurs when solder systems are used at these junctions. Intermetalic compounds are formed which crack under the stress created by dissimilar expansion coefficients of the joined parts. Because solder systems represent the most economical approach to package assembly, much work is being concentrated on developing a solder technique capable of withstanding continuous thermal cycling.

Thermal coupling on a monolithic chip is both a help and a hindrance. The close proximity of devices provides tight temperature tracking of parameters and this can be a distinct advantage in many areas of regulator design. Unfortunately, thermal coupling between the reference and the pass transistors forces the designer to allow for reference temperature changes whenever there is a shift in power dissipation in the pass transistors. This power is given by $P = (V_{IN}-V_{OUT})(I_{OUT})$. Changes in either load current or input voltage will affect power dissipation, and therefore the temperature of the pass transistor and the reference. The end result is that load and line regulation become at least partly dependent on the temperature stability of the reference.

In monolithic circuit designs which rely on very close thermal match between devices on a chip, temperature gradients may be a problem. This is especially true for high power monolithic regulators where the pass transistor area of the chip may be 5-50°C hotter than the rest of the chip. Thermal gradients between various components in the error amplifier or the reference will cause shifts in the output voltage of the regulator. To avoid this problem, critical components must be placed on isotherms. Because isotherms are very difficult to calculate, symmetry is used whenever possible. The problem is compounded by thermal phase delays such that two devices may be at nearly the same temperature in equilibrium, but not in the transient condition.

Thermal runaway is a well known phenomenon in discrete circuit designs where several power transistors are paralleled for higher current or power capability. Runaway occurs when the positive feedback effect of junction heating overcomes externally applied negative feedback. The external negative feedback is usually in the form of emitter degeneration applied equally but separately to the individual transistors. In monolithic designs, a large power transistor is broken up into several sections and each section individually ballasted to prevent thermal runaway within the power transistor itself. This sectioning provides added benefits in current sharing and collector resistance. In most cases, the ballasting resistors double as current limit sense resistors.

HIGH CURRENT EFFECTS

Electromigration is a troublesome effect occurring in aluminum interconnects where actual physical migration of aluminum occurs under high current density conditions. This phenomenon occurs at current densities well below that required for conventional fusing and is the limiting factor in the calculation of continuous current handling capability for an interconnect. Many factors such as grain size, impurity content and glassivation affect electromigration but allowable densities fall into the 10^5 to 10^6 A/CM^2 range. In the case of high current monolithic regulators, chip size may have to be increased simply to allow enough room for the large interconnects used on the emitter and collector of the pass transistor. A 10 amp regulator for instance would require aluminum interconnects 100 mils wide on collector and emitter of the pass transistor. Addition of impurities such as aluminum oxide and a protective covering of deposited silicon dioxide over the aluminum may offer significant reductions in the size of interconnects. Increasing thickness helps too, but etching problems restrict the thickness to several microns.

High current monolithic regulators have in recent years exceeded the current handling capability of the bonding wires used to connect the chip to the package leads. A standard one mil aluminum bonding wire fuses at about 1 amp. The National LM109 regulator will deliver up to 3 amps into a short circuit. Unfortunately fusing current is proportional to wire diameter, not area, so even the larger 3 mil bonding wire with a 3 amp fuse current is not sufficient for reliable use in the LM109. This necessitates the use of multiple bonds with attendent increase in chip area and added assembly cost. Although bonding wires up to 10 mils in diameter are feasible, the large bonding pads required for this size wire require an excessive amount of chip area.

Bonding wire resistance can be critical in 3 lead monolithic regulators because there is no provision for Kelvin sensing at the load. The resistance of the output bonding wires adds directly to the output impedance of the regulator. For example, the output impedance of the LM109 on the chip is typically 0.005Ω, but the resistance of the output bonding wires is 0.03Ω, giving a total output impedance seven times higher than would be obtained with Kelvin sensing. Performance on output impedance and load regulation, then, is limited by the package.

Resistance of bonding wires can also affect the maximum obtainable efficiency of a regulator. In a 3 amp regulator with two 3 mil bonding wires on the input and the output, 0.6W is consumed in the wires alone. A good regulator design will allow an input-output voltage differential of 2V minimum, so the 0.6W represents a 10% increase in wasted power for maximally efficient conditions.

STABILITY

Closed loop stability is of major concern in general purpose monolithic regulators because the external impedance seen by the output of the regulator can vary over such a wide range. The regulator loop must be stable for all load conditions including no-load to full-load current and a wide range of load capacitance. In a three-lead design, loop stabilization must be accomplished on the monolithic chip. Designs must not require more than about 50pF of capacitance because of the large chip area consumed by capacitors. Capacitance multiplier techniques are often used to make the effective value of the capacitor seem much larger than it is.

One new technique developed at National is particularly suited to two-stage negative regulators where the pass transistor is operated in the common emitter connection. The current reflector load for the first stage is also used as a capacitance multiplier with a gain of about 20. In addition, the circuit forms both a pole and a zero such that the first stage gain falls to unity independent of the operating current of the stage.

REFERENCE

Reference voltage drift with time is a problem in monolithic regulators. The reference is usually derived from an internal zener which has surface limited breakdown. Variations in surface conditions can affect the zener voltage. Specifically, mobile ions on or near the surface will migrate with time, shifting the breakdown voltage as much as 2%.

Commercially available reference zeners do not suffer from this effect in general because the junction has been etched away at the surface, forcing breakdown to occur in the protected subsurface region.

A new voltage reference circuit developed at National Semiconductor utilizes resistors and transistors operated in their active region only. No avalanche or zener mechanism is involved. The new reference has low noise, very good long term stability and is easily fabricated with normal monolithic components. Absolute tolerance of the reference voltage is typically ±3%.

The new reference makes use of the change in temperature dependence of the base-emitter voltage of a bipolar transistor as its emitter current density is varied. The base-emitter voltage difference between two transistors operated at current density J_1 and J_2 is given by

$$V_{BE1} - V_{BE2} = \frac{KT}{q} \ln \frac{J_1}{J_2} \qquad \begin{array}{l} K = \text{Boltzman's constant} \\ T = \text{Absolute temperature} \\ q = \text{Charge of electron} \end{array} \qquad (1)$$

This voltage difference is seen to be proportional to temperature. Differentiating (1) with respect to temperature yields:

$$\frac{d\left(V_{BE1} - V_{BE2}\right)}{dT} = \frac{K}{q} \ln \frac{J_1}{J_2} .$$

For $\frac{J_1}{J_2}$ equal to ten, the difference voltage is 60mV at 25°C and rises to 80mV at 125°C.

A zero temperature coefficient reference is made by multiplying this voltage difference to 600mV with appropriate circuitry and combining the 600mV with the base-emitter voltage drop of a transistor. A silicon bipolar transistor has a base-emitter temperature dependence very reproducible at about 2mV/°C. Assuming a base-emitter voltage of 600mV at 25°C, the value at 125°C will be 400mV. The 600mV obtained from the multiplied difference voltage will shift to 800mV at 125°C. The sum of these two components is 1.2V at 25°C and also 1.2V at 125°C, yielding a zero temperature coefficient.

Reference voltages higher than 1.2V can be obtained by amplifying the voltage difference of the two transistors to N(600mV) where N is an integer. N base-emitter junctions are then added in series with the amplified difference voltage to obtain a reference voltage of 2.4V, 3.6V, 4.8V, etc.

There are many advantages to the new voltage reference; the reference voltage is independent of minor variations in base or emitter dopings, the reference can be operated off lower voltage supplies than the 7-9 volts required for conventional zener references, and the reference can be used as its own error amplifier in regulator applications.

MATCHING

Matching between adjacent devices on a monolithic chip has been utilized by circuit designers for some time. Although many unusual and interesting applications of matched devices are known and used, this discussion will be limited to those applications particularly suited to regulator design.

Perhaps the most common use of a matched pair of transistors is in the differential stage of the error amplifier and as an active load for that stage. A matched transistor active load accomplishes four things; it balances the differential stage for good drift characteristics, it eliminates large resistor values in low current stages, it provides differential-to-single ended conversion and it raises the single stage voltage gain from less than 100 to more than 1000. This improvement in gain allows the use of a single stage error amplifier which is easy to stabilize under closed loop conditions with a minimum of capacitance.

Other uses of matched transistors in monolithic regulators include; active current sources, low sense voltage current limiters and active voltage reference circuits.

Matching between adjacent resistors on a monolithic chip with a normalized standard deviation of less than 1% is readily achieved. This means that the output divider used on a monolithic regulator does not contribute significantly to the spread in output voltage of the regulators. Of equal significance is the extremely good match in the temperature coefficient of the resistors so that divider ratios hold over the operating temperature range of the regulator.

CURRENT LIMITERS

A more recent use of the $V_{BE}/\Delta V_{BE}$ reference technique is in current limiter circuits which have a current limit sense voltage of 50 to 200mV. This low sense voltage increases regulator efficiency and can be made with any desired temperature coefficient between +0.3%/°C and -0.3%/°C. Foldback limiting or power limiting may be added if desired.

The technique for obtaining the current limit sense voltage is as follows: A base-emitter difference voltage is obtained by operating two transistors at different current densities. This difference voltage is added directly to a fraction of a base-emitter voltage drop to obtain the current limit sense voltage. The temperature coefficient may be varied by changing the relative magnitude of the ΔV_{BE} portion and the fractional V_{BE} portion. The ΔV_{BE} voltage has a +0.3%/°C temperature dependence, while the fractional V_{BE} has a -0.3%/°C temperature dependence.

TEMPERATURE AND POWER LIMITERS

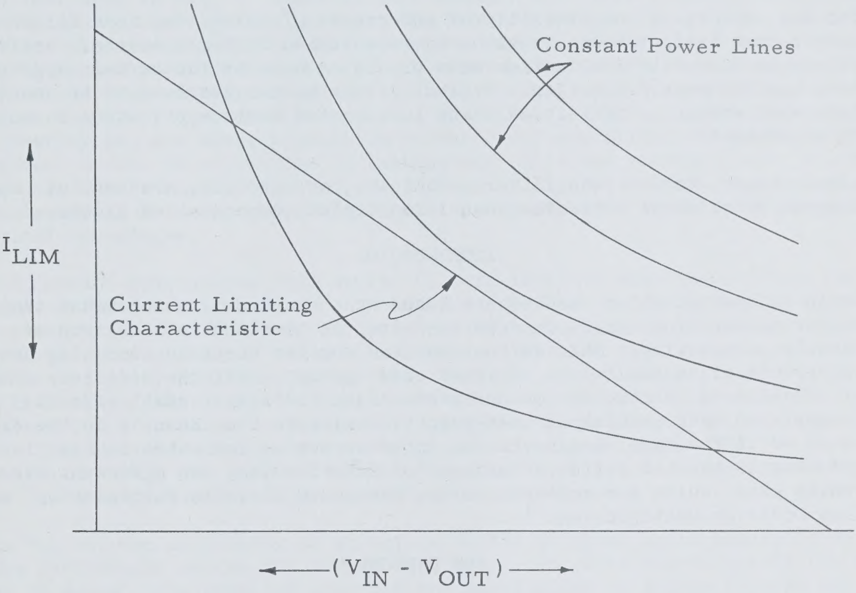
Monolithic voltage regulators must be designed to work under a wide variety of operating conditions. Unknowns include input operating voltage, exact load conditions and the thermal resistance of any additional heat sink used. In an effort to make monolithic regulators "goof-proof" and still retain good power handling capability, temperature limiting has been added to many designs. The technique is simply to sense chip temperature and shut off the output current when the chip temperature rises above a predetermined value - typically 150 to 175°C.

The sensor used to detect chip temperature is usually a transistor with a fixed base-emitter voltage applied. This voltage is about 400mV so that the transistor is off at low temperatures, but turns on at a predetermined high temperature, initiating shutdown of the regulator.

The pass transistor (heat source) and the thermal shutdown transistor (temperature sensor) form a thermal feedback loop. Thermal phase delays across the chip between the heater and sensor can cause loop oscillations. Oscillation frequency can be in the range of 1Hz to 10kHz depending on loop gain and heater to sensor spacing. Hysteresis is often added deliberately to force oscillation frequency down. The oscillation is not harmful in most cases because average chip temperature is maintained within a few degrees of the desired temperature.

One problem with thermal shutdown is that the thermal phase delay between the pass transistor and the sensor may be long enough to let pass transistor temperature reach destructively high levels before the sensor reacts. This occurs when the chip is initially at a low temperature and a step function of power is created in the pass transistor. The temperature of the pass transistor may reach 300-400°C before the sensor temperature rises to the shut off limit of 150 to 175°C. After equilibrium is reached, the pass transistor temperature might be only a few degrees above the sensor temperature. The initial surge in temperature, however, is often fatal to the regulator and is evidenced by melted interconnects, fused bonding wires and sometimes pieces literally blown off the chip.

To overcome the phase delay problem, power limiting can be added to the regulator. This is done by using current limiting modified by input-output voltage differential. A voltage proportional to the difference between the input and output voltage of the regulator is added to the voltage developed across the current limit sense resistor. The maximum power created in the pass transistor is thereby limited to a safe value regardless of input voltage or output shorts. The graph below shows the effect of adding power limiting to a regulator.



By proper choice of initial current limit value and slope of the current limit versus $(V_{IN}-V_{OUT})$, the regulator will be limited to a reasonably fixed power over most of its operating range.

ANALOG AND DIGITAL FILTERING IN MULTIPLEX COMMUNICATION SYSTEMS

CARL F. KURTH
Bell Telephone Laboratories, Incorporated
North Andover, Massachusetts 01845

ABSTRACT

The design and realization of filters in communication systems have been influenced by various technologies over the past years. Beyond the traditional LC-filter technology, a sophisticated crystal filter and, more recently, a mechanical filter technology have been developed. Each filter technology is applied in multiplex communication systems according to its inherent technical and economical advantages.

With the economical acceptability of integrated circuits, two more filter technologies begin to compete with the established filter concepts. Active filters in thin film realization show promising aspects for certain applications and for mass production. Digital filter techniques seem to be understood well enough so that it is worth looking for their application in multiplex systems.

In this paper, the various filter techniques, new and old, are set into the perspective of their economical use in multiplex communication systems.

INTRODUCTION

Multiplex communication systems are based on a hierarchy, which means that a smaller number of channels on separate wires is combined into a group of channels on one wire. This is continued in further steps by combining several groups of channels into a higher order group, until the necessary number of channels is combined on one wire. Each individual combination is accomplished by a modulation step which interleaves the channels in the frequency or in the time domain without interference as indicated in Fig. 1. According to the two different methods of interleaving, two different hierarchies exist which are referred to as "Frequency Division Multiplexing" and "Time Division Multiplexing."[1]

FDM SYSTEMS

Frequency division multiplexing (FDM) is based on amplitude modulation (AM) and linear filtering where only one sideband is transmitted. Consequently, this is the area where filters are needed, on the transmitting side to separate the unwanted sideband, or on the receiving side to separate the individual channels before demodulation as shown in Fig. 2. In modern FDM communication systems, the channels are staggered over the frequency band in 4-kHz slots, thus, the filters for separating the individual channels have relatively steep attenuation slopes since, within each 4-kHz slot, a bandwidth of 3.2 kHz is used for transmission. One of these channel filters is necessary in the transmitter and another in the receiver; therefore, the demand is four times the number of communication channels installed annually. Hence, the production volume of those filters is high, which results in a rather cost sensitive design.[2,3,5,6]

The channel filters have an international history since the first step of modulation in FDM systems is internationally standardized. According to the most economical technology available, these filters have been designed as LC filters, crystal filters, monolithic crystal filters, and mechanical filters. Price and manufacturability at a given time have influenced the technology which is used in the various countries. Each of the above-mentioned filter technologies has its own preferred frequency range in which the filters were realized for best technical performance. This, in turn, influenced the system concept, such that with one filter design, there is only one step of modulation, for another design two steps of modulation are necessary to place a group of 12 channels into a prescribed frequency band of 60 to 108 kHz. In all cases, the resulting transmission performance is compatible. In Fig. 3 and 4, two typical modulation schemes are shown for generating a group of 12 single sideband channels.[4,7,8,9]

While the channel filters have to provide the high selectivity for separating the individual channels, the filters in the following modulation steps are only necessary to suppress unwanted modulation products which might fall into frequency bands that are reserved for further use with other channels. As indicated in Fig. 5, those filters usually have less stringent requirements on the selectivity; however, the requirements on the linearity for the filter plus modulator are rather severe so that channel interference introduced by nonlinear distortion is kept within limits. The more channels combined and transmitted through one filter, the higher is the relative bandwidth and the frequency allocation of the filter. Thus, the requirements on linearity and the lower production volume for these filters have so far prevented the use of other than LC techniques. With the present state of the art, it can be assumed that active filters or digital filters may be used successfully in a channel filter application, provided that it yields economical advantages.

The present experiences with active filters indicate that these filter concepts influence even more the systems concept than the filter technologies previously used. Active RC filters have to be designed in frequency ranges and with configurations which allow them to meet the requirements on temperature stability and which assure that the design results in an unconditionally stable filter despite the phase shift and roll off in the active device. The present thin film technology raises high hopes that, at least in the frequency band below 50 kHz, high quality active filters will be competitive, although the dc power drain of an active filter is a matter which still causes raised eyebrows by the systems designer..

In Fig. 6, the complexity of an active filter is shown which approximates the performance needed for a channel filter. It draws approximately 200 mW of dc power. The pros and cons for the application of active filters are listed in Fig. 7. With the present state of the technology, on a technical plus economical basis, it is apparent that active filters have a chance to compete in the lowest level of the hierarchy, but even there, they have to meet tough competition in the well established passive design.[10,11,12,13]

The application of digital filters will influence the system design severely. As demonstrated in Fig. 8, the digital filter can process only digital signals; the information has to be converted first from an analog format into a digital one, and then the digital filtering can be performed.[15] Although features such as unconditional stability, high precision, multiplexibility, and integratability are attractions solely offered by digital filters, their applicability for filtering in FDM systems will depend very much on the success of building inexpensive A/D and D/A converters. Furthermore, the dc power drain is an even greater obstacle, since the power drain is almost proportional to the speed of the digital hardware. Besides these negative aspects which might be overcome in the course of further development work, digital filters are attractive since large scale integration can be applied readily which has a great potential for cost reduction.[17]

Due to the quantized representation of the signal in the digital filter, noise is introduced which is related to the number of quantization steps over the total dynamic range or to the finite number of bits per signal sample. In order to achieve acceptable signal-to-noise ratios in an FDM communication system, the word length in a digital filter may have to be as high as 16 bits or more. This, in turn, results in relatively high bit rates, particularly if the filter is multiplexed over several channels. The bit rate of the filter is also proportional to the sampling rate which, in turn, has to be higher than the highest frequency transmitted in order to obey Nyquist's theorem. Consequently, the frequency range in which digital filters can be applied is determined by the speed of the available hardware. With the present state of the art for digital hardware, the application of digital filters in FDM systems is limited to the hierarchy level of the channel filter, particularly under the aspect of multiplexing the digital filter over 12 channels for economical advantages.

More recent experiences with digital filters indicate that a digital implementation of the "Weaver Modulator" is a possible way for single sideband generation.[16,17,18] The "Weaver Modulator" is a double quadrature modulator with two signal paths. At the end of the two signal paths, the unwanted sideband is canceled by adding or subtracting the two output signals. Previous experiences with this approach suffered severely from the limited accuracy of an analog implementation of the two signal paths. This resulted in a limited suppression of the unwanted sideband such that this approach never got much attention for commercial use in FDM systems. The digital implementation overcomes this shortcoming since the amount of suppression is solely related to the number of bits per word. A system implemented with a word length which is sufficient for a required signal-to-noise ratio will also yield sufficient suppression of the unwanted sideband. In Fig. 9, a block diagram of a digital "Weaver Modulator" is shown which is timeshared over K channels.

The type of quantization in the multipliers within the digital filters will have an influence on the amount of additional noise introduced into the channel. It also has an influence on the "limit cycle" behavior of the filters. Recent investigations indicate that sign-magnitude truncation has an advantage over a straightforward round off procedure, particularly in the realization of digital bandpass filters.[19] This could have some bearing on the choice of the system concept.

TDM SYSTEMS

Time division multiplexing (TDM) is based on sampling and encoding the signals (A/D)conversion) of several channels which then are interleaved in the time domain without interference.[20,21] A typical example is shown in Fig. 10. Contrary to the FDM method, the separation between the different channels is here accomplished by allocating a certain time slot to each channel and maintaining synchronism between transmitter and receiver. In the A/D and D/A conversion, it has to be assured that the energy per signal sample before and after encoding is confined to the appropriate time slot, otherwise, intersymbol interference would lead to crosstalk. As can be seen, the separation of the channels is here accomplished without any filtering at all. The only filtering which is necessary in a TDM system is before and after the A/D and D/A conversion. On the transmitting side, the spectrum of the signal which is being sampled has to be bandlimited. According to Nyquist's sampling theorem, the sampling rate has to be higher than the highest frequency in the spectrum to be sampled. The higher the sampling rate, the less expensive is the filtering. However, A/D converters become more expensive the higher the sampling rate is. So far in most voice frequency TDM systems, a sampling rate of 8 kHz has been used which resulted in the need of a fourth- or fifth-order filter before A/D conversion. At the receiving side, a filter is necessary for integrating the signal samples per channel after the D/A conversion.

Both filters are typical candidates for an active implementation. Usually, the size reduction which comes automatically with an active filter is welcomed in an all digital system. In addition, the sampling on the transmitting side can be combined advantageously with the active filter.[22,23] Finally, the active filter can provide the gain, which is necessary to adjust the final level at the input and output of the TDM system. The example shown in Fig. 11 indicates the principle of A/D conversion by using an active filter.

Future development of inexpensive A/D converters with a high sampling rate may reduce the requirements on these filters or even eliminate them by introducing a digital filter after A/D conversion as shown by the example in Fig. 12. This seems to introduce more complexity since the companding, typical for TDM systems, is usually combined with the A/D conversion. Hence, the digital filtering would have to be done with a relatively high bit rate before companding. To what extent under these circumstances the digital filter can compete successfully with an analog filter is a matter of future development in the field of digital hardware.[24]

CONCLUSION

It seems fair to conclude that neither the active nor the digital filter technology can presently compete against passive LC-designs in the higher levels of the FDM hierarchy, those above 200 kHz.

At the lowest level of the FDM hierarchy, the active and digital filter approaches are competing against a recently established, highly promising crystal or mechanical filter technology, which is about to be introduced into the channel bank of FDM systems. Although the technical feasibility of active filters for the channel bank application has been established, economic advantages and outstanding features with respect to the system's concept have not yet been proven.

The introduction of digital filters into the lowest level of the hierarchy of either FDM or TDM systems depends very much on the further development of inexpensive, high speed digital hardware with low power consumption. In addition, the necessary A/D and D/A converters with a higher sampling rate should not be overlooked in the cost picture, particularly when the digital filters have to compete against well established system concepts based on analog filter designs.

The use of active filters in RC thin film technology for bandlimiting the voice signals in PCM systems appears to be the first breakthrough of this technology in mass production. This was possible since the RC thin film technology, together with the active filter approach, naturally allow the application of automated manufacturing processes which, in turn, influence the economics considerably.

As a final conclusion, it can be said that the various filtering techniques are well understood. Their successful introduction into multiplex communication systems depends, however, mainly on the development of economical manufacturing processes. In other words, the filters should be designed such that they can be manufactured economically on a large production scale. This, in turn, is very much influenced by the annual production volume which indicates that, to a great extent, demand sets the pace for introducing new technologies,

ACKNOWLEDGMENT

On a subject of this diversity, it is difficult to acknowledge individually the many discussions the author had throughout his career with colleagues in the field of networks and communication technology, which led to the present understanding of the various aspects of filtering methods in multiplex systems. Suffice it to say that the help they have given in arriving at the ideas presented is fully appreciated.

REFERENCES

1. Bell Telephone Laboratories, "Transmission Systems for Communications," Fourth Edition, February 1970.

2. F. H. Blecher, "The Transistorized A5 Channel Bank for Broadband Systems," B.S.T.J., Vol. 41, January 1962.

3. K. Endo et al, "New Channel Translating Equipment," Fujitsu Scientific and Technical Journal, Vol. 3, No. 1, March 1967.

4. S. W. Anders, "A Polylithic Filter Channel Bank," IEEE Trans. on Communications, Vol. COM-20, No. 1, February 1972.

5. A. Reading, R. J. Martin, "A New Modulation Concept in Channel Translating Equipment," IEEE Trans. on Communications, Vol. COM-20, No. 1, February 1972.

6. G. W. Bleisch, "The A6 Channel Bank," IEEE Trans. on Communications, Vol. COM-20, No. 2, March 1972.

7. P. Lloyd, "Monolithic Crystal Filters for Frequency Division Multiplex," Conf. Proc. of the 25th Symposium on Frequency Control, Atlantic City, April 1971.

8. H. Albsmeier, "Ein Vergleich der Realisierungsmoeglichkeiten elektromechanischer Kanal-filter im Frequenzbereich 12 kHz bis 10 MHz," Frequenz, Vol. 25, No. 3, 1971.

9. M. Boerner, E. Kettel, and H. Ohnsorge, "Mechanische Filter fuer die Nachrichtentechnik," Telefunken Zeitung, Vol. 31, June 1958.

10. G. S. Moschytz, "Inductorless Filters - A Survey," Proc. of 20th Electronic Components Conference, Washington, D.C., May 1970.

11. S. K. Mitra, Active Inductorless Filters, IEEE Press, New York, 1971.

12. F. J. Witt, Personal Conversation.

13. G. S. Moschytz, W. Thelen, "Design of Hybrid Integrated Filter - Building Blocks," IEEE Journal of Solid State Circuits, Vol. SC-5, No. 3, June 1970.

14. R. A. Friedenson, "Computer Aided Design of Practical RC-Active Filters," IEEE International Symposium of Circuit Theory, Los Angeles, Cal., April 1972.

15. L. B. Jackson, J. F. Kaiser, and H. S. McDonald, "An Approach to the Implementation of Digital Filters," IEEE Trans. Audio Electroacoust., Vol. AU-16, September 1968.

16. D. K. Weaver, Jr., "A Third Method of Generation and Detection of Single Sideband Signals," Proc. IRE, Vol. 44, December 1956.

17. S. L. Freeny, R. B. Kieburtz, K. V. Mina, and S. K. Tewksbury, "Design of Digital Filters for an All Digital Frequency Division Multiplex-Time Division Multiplex Translator," IEEE Trans. Circuit Theory, Vol. CT-18, No. 6, November 1971. "Systems Analysis of a TDM-FDM Translator/-Digital A-type Channel Bank," IEEE Trans. on Communication Technology, Vol. COM-19, No. 6, December 1971.

18. C. F. Kurth, "SSB/FDM Utilizing TDM Digital Filters," IEEE Trans. on Communication Technology, Vol. COM-19, No. 1, February 1971.

19. C-Y. Kao, "An Analysis of Limit Cycles Due to Sign-Magnitude Truncation in Multiplication in Recursive Digital Filters," 5th Asilomar Conference on Circuits and Systems, Pacific Grove, Cal., November 1971.

20. H. H. Henning, "A 96-Channel PCM Channel Bank," Conference Record, IEEE International Conference on Communications, 1969.

21. K. W. Cattermole, "Efficiency and Reciprocity in Pulse-Amplitude Modulation," Proc. IEEE, Vol. 105, Part B, 1958.

22. M. L. Liou, P. H. McDonald, and W. Thelen, "Computer Optimization of Active and Passive Switched Low-pass Filters for PCM," Proc. of the 1970 Electronics Components Conference, May 1970.

23. R. L. Adams, "On Reduced Sensitivity Active Filters," Proc. of the 14th Midwest Symposium on Circuit Theory, May 1971.

24. J. F. Kaiser, "An Overview of Digital Filters," IEEE Newsletter, Circuit Theory Group, Vol. 6, No. 1, March 1972.

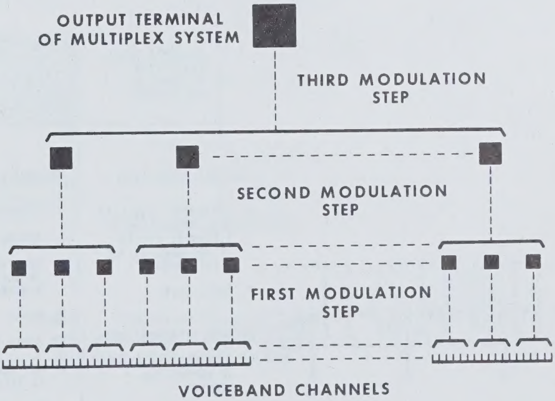


FIG.1 — HIERARCHY OF MULTIPLEX COMMUNICATION SYSTEM

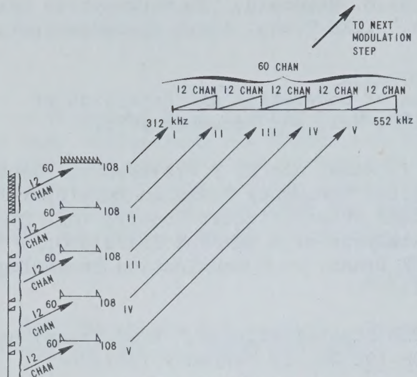

TO NEXT MODULATION STEP
60 CHAN
12 CHAN
12 CHAN
12 CHAN
12 CHAN
12 CHAN
312 kHz
552 kHz
I
II
III
IV
V
60
108
12 CHAN
0-4 kHz

FIG.2 — FREQUENCY DIVISION MULTIPLEX HIERARCHY

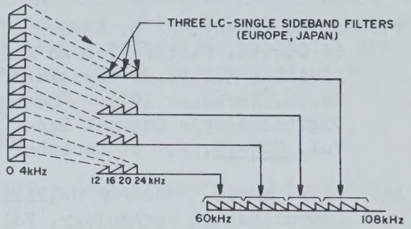

THREE LC-SINGLE SIDEBAND FILTERS (EUROPE, JAPAN)
0 4kHz
12 16 20 24 kHz
60kHz
108kHz

FIG. 3 — SINGLE SIDEBAND CHANNEL BANK WITH PREMODULATION BETWEEN 12 AND 24kHz

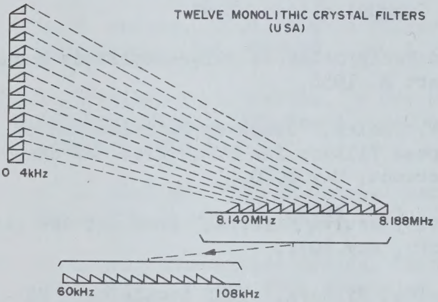

TWELVE MONOLITHIC CRYSTAL FILTERS (USA)
0 4kHz
8.140MHz
8.188MHz
60kHz
108kHz

FIG.4 — SINGLE SIDEBAND CHANNEL BANK WITH PREMODULATION BETWEEN 8.140 AND 8.188MHz

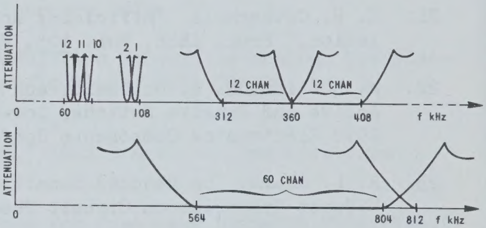


FIG.5 — FILTERS IN FDM-HIERARCHY

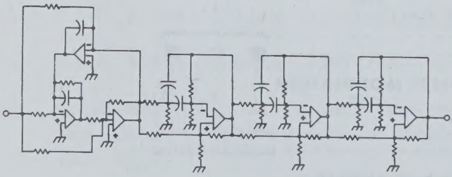
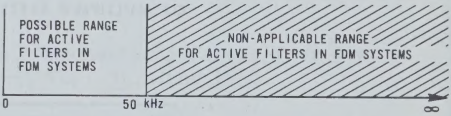


FIG. 6 — COMPLEXITY OF ACTIVE FILTER WHICH APPROXIMATES CHANNEL FILTER REQUIREMENTS


POSSIBLE RANGE FOR ACTIVE FILTERS IN FDM SYSTEMS
NON-APPLICABLE RANGE FOR ACTIVE FILTERS IN FDM SYSTEMS
0
50 kHz
∞

POSSIBLE ADVANTAGES

1. AUTOMATED THIN FILM MASS PRODUCTION OF A CHANNEL FILTER IF DESIGNED AT THE RIGHT FREQUENCY.
2. SMALL SIZE.

PRESENT DISADVANTAGES

1. NECESSITATES CHANGE IN MODULATION CONCEPT.
2. PRICE?

PRESENT DISADVANTAGES

1. STABILITY VERSUS TIME AND TEMP.?
2. ACTIVE DEVICES ROLL OFF.
3. NONLINEAR DISTORTION REQUIREMENTS IN MULTIPLEXED CHANNELS TOO STRINGENT FOR THE ACTIVE DEVICES.
4. NOISE.
5. POWER CONSUMPTION.
6. NOT ECONOMICALLY COMPETITIVE WITH LC FILTERS AT HIGH FREQUENCIES.
7. LOW PRODUCTION VOLUME OF FILTERS IN HIGHER ORDER FDM HIERARCHY

FIG. 7 — APPLICATION OF ACTIVE FILTERS IN FDM-SYSTEMS

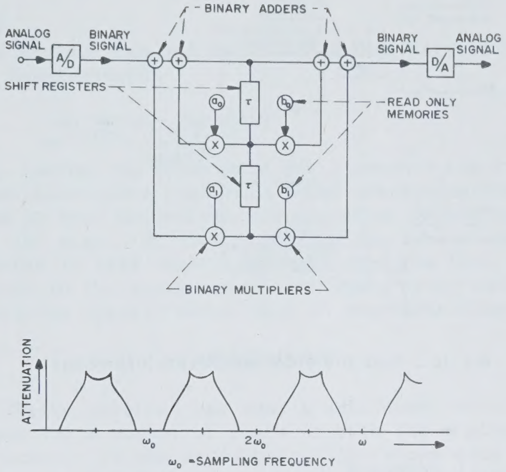

BINARY ADDERS
ANALOG SIGNAL
A/D
BINARY SIGNAL
SHIFT REGISTERS
τ
a_0
b_0
a_1
b_1
READ ONLY MEMORIES
BINARY MULTIPLIERS
BINARY SIGNAL
D/A
ANALOG SIGNAL
ATTENUATION
ω_0
$2\omega_0$
ω
ω_0 = SAMPLING FREQUENCY

FIG. 8 — DIGITAL FILTER OPERATED BETWEEN ANALOG-DIGITAL CONVERTERS AND TYPICAL FREQUENCY RESPONSE

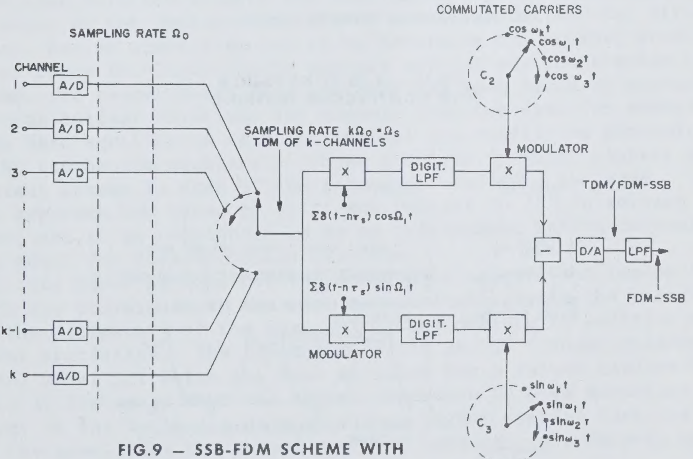

SAMPLING RATE Ω_0
CHANNEL
1
2
3
k-1
k
A/D
SAMPLING RATE $k\Omega_0 = \Omega_s$
TDM OF k-CHANNELS
X
DIGIT. LPF
$\Sigma\delta(t-n\tau_s)\cos\Omega_1 t$
$\Sigma\delta(t-n\tau_s)\sin\Omega_1 t$
MODULATOR
COMMUTATED CARRIERS
$\cos\omega_k t$
$\cos\omega_1 t$
$\cos\omega_2 t$
$\cos\omega_3 t$
C_2
MODULATOR
TDM/FDM-SSB
−
D/A
LPF
FDM-SSB
$\sin\omega_k t$
$\sin\omega_1 t$
$\sin\omega_2 t$
$\sin\omega_3 t$
C_3

FIG.9 — SSB-FDM SCHEME WITH WEAVER MODULATOR AND DIGITAL LOW-PASS FILTER

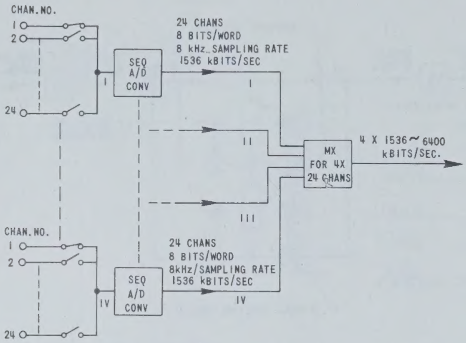
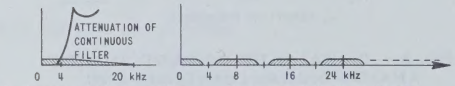
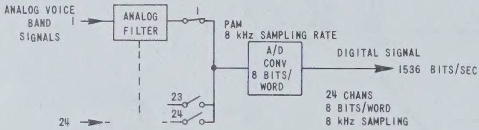

CHAN. NO.
1
2
24
SEQ A/D CONV
I
24 CHANS
8 BITS/WORD
8 kHz. SAMPLING RATE
1536 kBITS/SEC
I
II
III
IV
MX FOR 4X 24 CHANS
4 X 1536 ~ 6400 kBITS/SEC.
CHAN. NO.
1
2
24
SEQ A/D CONV
IV
24 CHANS
8 BITS/WORD
8kHz/SAMPLING RATE
1536 kBITS/SEC

FIG. 10 — TIME DIVISION MULTIPLEX HIERARCHY


ATTENUATION OF CONTINUOUS FILTER
0
4
20 kHz
0
4
8
16
24 kHz

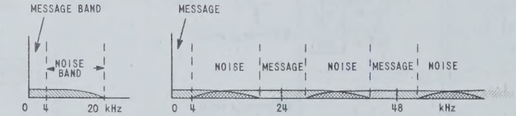
A. BAND LIMITATION BEFORE SAMPLING

B. SPECTRUM AFTER SAMPLING AT 8 kHz RATE.


ANALOG VOICE BAND SIGNALS
1
ANALOG FILTER
1
PAM
8 kHz SAMPLING RATE
A/D CONV 8 BITS/ WORD
DIGITAL SIGNAL
1536 BITS/SEC
24 CHANS
8 BITS/WORD
8 kHz SAMPLING
23
24
24

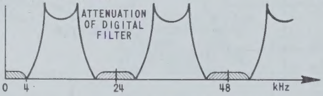
C. BLOCK DIAGRAM OF MODULATION PRINCIPLE.

FIG.11 — A/D-CONVERSION WITH CONTINUOUS FILTERING


MESSAGE BAND
NOISE BAND
0
4
20 kHz
MESSAGE
NOISE
MESSAGE
NOISE
MESSAGE
NOISE
0
4
24
48
kHz

A. UNLIMITED SPECTRUM BEFORE SAMPLING

B. UNLIMITED SPECTRUM AFTER SAMPLING AT 24 kHz RATE.


ATTENUATION OF DIGITAL FILTER
0
4
24
48
kHz

C. BAND LIMITED SPECTRUM AFTER DIGITAL FILTERING.

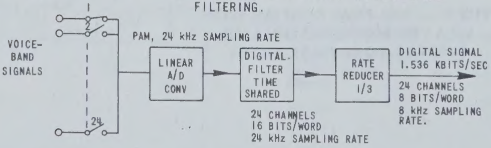

1
2
24
VOICE-BAND SIGNALS
PAM, 24 kHz SAMPLING RATE
LINEAR A/D CONV
DIGITAL FILTER TIME SHARED
RATE REDUCER 1/3
DIGITAL SIGNAL
1.536 KBITS/SEC
24 CHANNELS
8 BITS/WORD
8 kHz SAMPLING RATE.
24 CHANNELS
16 BITS/WORD
24 kHz SAMPLING RATE

FIG.12 — A/D-CONVERSION WITH DIGITAL FILTERING

A CONTRADICTORY PROPERTY OF THE NASH SOLUTION FOR TWO STOCHASTIC NONZERO-SUM GAMES*

TAMER BASAR
Division of Engineering and Applied Physics, Harvard University
Cambridge, Massachusetts

ABSTRACT

A desirable optimal solution pair for a nonzero-sum two-person game might be the one which gives rise to a lower average optimal cost for the player who seems to have unquestionable advantage on his co-player prior to the start of the game. We show that, for the scalar and single-stage problems considered in this paper, and employing the Nash solution concept, the LQG model used in the engineering literature never satisfies this property, whereas the duopoly model used in economics always does.

I. INTRODUCTION

Nash equilibrium solution has been a well-known solution concept for nonzero-sum games for a number of years in both the engineering and the economics literature. In the engineering literature some rather extensive work has been done on deterministic nonzero-sum differential games (see for example [1], [2]) with particular emphasis given on two-person games of the linear quadratic type in order to obtain explicit Nash policies (see for example [3]). It is not difficult to deduce from the current literature that if the players enter the game rather symmetrically, if they have the same facilities in computing their optimal policies, and further if they agree ahead of time to act rationally during the course of the game, then the Nash solution is in fact the only equilibrium solution that can be sought.

The same solution concept has been introduced in economics (there widely known as the Cournot equilibrium solution) for several different purposes. One of these purposes is to determine the optimal production levels of firms in oligopolistic markets without any cooperation between the firms. In recent years this approach has been taken by several authors in deriving optimal solutions for duopoly problems (see for example [4]).

The Nash equilibrium solution concept can readily be generalized to encompass stochastic problems -- those problems in which players do not have direct access to some of the parameters defining the game. In fact such an approach was taken by Cyert and DeGroot in [5] in solving duopoly problems, and it is considered to be an improvement on the deterministic Cournot model for some economics problems.

In this paper we consider two stochastic nonzero-sum two-person games in which the players enter the game symmetrically, with the exception of one of the parameters of the game which they measure in additive noise of different statistics. The first problem is an LQG (linear quadratic gaussian) model for which the Nash solution has a rather counter-intuitive property in the sense that the player who seems to have advantage on his co-player at the beginning ends up with a higher average Nash cost at the end of the game. For the second problem, however, -- a duopoly model --

---

*This work was partially supported by the National Science Foundation Grant GK 31511 and the U.S. Office of Naval Research under the Joint Services Electronics Program by Contract N00014-67-A-0298-0006 through the Division of Engineering and Applied Physics, Harvard University.

the Nash solution has a property which consistently contradicts with that of the first problem.

This conflict in the properties of the Nash solutions of two, what seems to be conceptually similar, game problems brings up the question of whether the Nash solution is the "best" solution concept for all such games, since the above mentioned counter-intuitive property might not be desired in seeking an optimal solution for the problem at hand.

II. PROBLEM 1

The first problem to be considered in this paper is a nonzero-sum two-person stochastic game defined by the quadratic cost functions

$$J_1(u,v) = (x+u+v)^2c + u^2d \tag{1a}$$

$$J_2(u,v) = (x+u+v)^2c + v^2d, \tag{1b}$$

where $c > 0$, $d > 0$ are scalars known to both players, and $x \in N[0,\sigma]$, $\sigma > 0$. Players 1 and 2 observe the random variable x through the noisy observations z_1 and z_2 respectively, where

$$z_1 = x + w_1, \qquad w_1 \in N[0,s_1],\ s_1 > 0 \tag{2a}$$

$$z_2 = x + w_2, \qquad w_2 \in N[0,s_2],\ s_2 > 0 \tag{2b}$$

with w_1, w_2 and x being statistically independent and $s_2 > s_1$. This last inequality roughly implies that Player 2 has "more observation noise" than Player 1.

Player 1 picks u as a measurable function of his observation z_1, and similarly Player 2 picks v as a measurable function of his observation z_2. Denoting the space of all Borel measurable functions mapping $\mathcal{R}^1$ into $\mathcal{R}^1$ by Σ, we define the average cost function $\bar{J}_i(u,v)$ for the i'th player, for fixed $u(.)$, $v(.) \in \Sigma$, by

$$\bar{J}_i(u,v) = E\{J_i(u(z_1),v(z_2))|u(.),v(.)\},\ i=1,2, \tag{3}$$

where $E\{.\}$ denotes the expectation operation taken over the prior statistics of the random variables x, w_1 and w_2.

Definition of a Nash Solution:

For the stochastic game posed above, the pair $\{u^*(.) \in \Sigma,\ v^*(.) \in \Sigma\}$ is said to be a Nash solution pair if for all $u(.) \in \Sigma$, $v(.) \in \Sigma$, the following relations are satisfied:

$$\bar{J}_1(u^*,v^*) \leq \bar{J}_1(u,v^*) \tag{4a}$$

$$\bar{J}_2(u^*,v^*) \leq \bar{J}_2(u^*,v). \tag{4b}$$

$\bar{J}_i(u^*,v^*)$ is said to be the expected (average) Nash cost of Player i.

Now, it is intuitively obvious that if $s_1 = s_2$ in the formulation of the LQG game problem posed above, then the players enter the game under absolutely symmetric (identical) situations and hence should face the same expected Nash cost, i.e. $\bar{J}_1(u^*,v^*) = \bar{J}_2(u^*,v^*)$ when $s_1 = s_2$, for all $c, d, \sigma > 0$.

However, when $s_1 < s_2$ the symmetry is broken since, even though the two cost functions (1a) and (1b) have the same symmetric form, Player 1 now has an observation noise with smaller variance than the one of Player 2. A limiting case of this situation would be the case when $s_1 = 0$, $s_2 = +\infty$, i.e. when Player 1 observes x perfectly and Player 2 has access to no observation. Since the prime goal of the i'th player is to minimize his own expected cost which is comprised of the same terminal cost for both players and a second term involving the cost of control to the i'th player -- assuming that the j'th player ($j \neq i$) acts rationally -- it would seem true at first glance that the player who has "more observation noise" should end up facing a higher expected Nash cost. However, we will show in the sequel that this conclusion can never be drawn for Problem 1:

Proposition 1:

The Nash solution pair for Problem 1 is given for all c, d, s_1, s_2, $\sigma > 0$ by

$$u^*(z_1) = -\left\{1 \Big/ \left[1 - \left(\frac{c}{c+d}\right)^2 \left(\frac{\sigma}{\sigma+s_1}\right)\left(\frac{\sigma}{\sigma+s_2}\right)\right]\right\} \cdot \left(\frac{c}{c+d}\right)\left(\frac{\sigma}{\sigma+s_1}\right)\left(1 - \frac{c}{c+d} \cdot \frac{\sigma}{\sigma+s_2}\right) z_1 \tag{5a}$$

$$v^*(z_2) = -\left\{1 \Big/ \left[1 - \left(\frac{c}{c+d}\right)^2 \left(\frac{\sigma}{\sigma+s_1}\right)\left(\frac{\sigma}{\sigma+s_2}\right)\right]\right\} \cdot \left(\frac{c}{c+d}\right)\left(\frac{\sigma}{\sigma+s_2}\right)\left(1 - \frac{c}{c+d} \cdot \frac{\sigma}{\sigma+s_1}\right) z_2 . \tag{5b}$$

Proof:

Minimizing (3) over $u(.) \in \Sigma$, for i=1 and for fixed $v(.) \in \Sigma$ we obtain

$$u = -\frac{c}{c+d}\left[\frac{\sigma}{\sigma+s_1} z_1 + E\{v|z_1\}\right]; \tag{6a}$$

and similarly, minimizing (3) over $v(.) \in \Sigma$ for i=2 and for fixed $u(.) \in \Sigma$ we get

$$v = -\frac{c}{c+d}\left[\frac{\sigma}{\sigma+s_2} z_2 + E\{u|z_2\}\right] . \tag{6b}$$

Now, it follows from (6a) and (6b) that the Nash policies u* and v* will be linear in z_1 and z_2 respectively; hence letting $v = \alpha z_2$, $u = \beta z_1$, we have

$$E\{v|z_1\} = \frac{\alpha\sigma}{\sigma+s_1} z_1 \tag{7a}$$

$$E\{u|z_2\} = \frac{\beta\sigma}{\sigma+s_2} z_2. \tag{7b}$$

Substituting (7a) and (7b) into (6a) and (6b) respectively, we obtain the equations

$$\begin{aligned} \beta &= -\left(\frac{c}{c+d}\right)\left(\frac{\sigma}{\sigma+s_1}\right)(1+\alpha) \\ \alpha &= -\left(\frac{c}{c+d}\right)\left(\frac{\sigma}{\sigma+s_2}\right)(1+\beta) \end{aligned} \tag{8}$$

and the solution pair to (8) determines u* and v* as given by (5a) and (5b). Q.E.D.

Property 1:

The Nash equilibrium solution pair (5a)-(5b) for Problem 1 has the counter-intuitive property that whenever $s_1 < s_2$, $\bar{J}_1(u^*,v^*) > \bar{J}_2(u^*,v^*)$ for all c, d, $\sigma > 0$.

Proof:

It follows from the definitions of $J_1(\cdot,\cdot)$ and $J_2(\cdot,\cdot)$ that

$$\bar{J}_1(u^*,v^*) - \bar{J}_2(u^*,v^*) = E\{(u^{*2}-v^{*2})d\}. \tag{9a}$$

After substituting for u*,v* given by (5a) - (5b) in (9a), some straightforward but rather extensive manipulations yield:

$$\bar{J}_1(u^*,v^*) - \bar{J}_2(u^*,v^*) = \kappa(s_2-s_1), \tag{9b}$$

where $\kappa > 0$ is defined by

$$\kappa = c^2d\sigma^2/[(c+d)^2(\sigma+s_1)(\sigma+s_2) - c^2\sigma^2]. \tag{10}$$

Property 1 now follows trivially from expression (9b) by observation, since $\kappa > 0$ for all c, d, $\sigma > 0$. Q.E.D.

Hence, roughly speaking, Property 1 says that Player 1 pays more for what seems to be his co-player's disadvantage in the game. For the extreme case when Player 1 observes x perfectly and Player 2 has access to no observation, it follows from (9b) and (10) that the average cost to Player 1 is more than the average cost to Player 2 by an amount $[(c/c+d)^2d\sigma]$.

III. PROBLEM 2

The second problem to be considered in this paper is a stochastic Cournot duopoly situation, i.e. the problem of two firms with identical products operating in a market in which the structure of the market demand function is known. To be more precise, we consider a linear demand curve of the form

$$p = \alpha - \beta(q+r), \tag{11}$$

where p denotes the price of the commodity, q is the production level of Firm 1, r is the production level of Firm 2, and (q+r) is the total quantity demanded. $\beta > 0$ is a scalar that determines the slope of the demand curve and it is known to both firms. α is a random variable whose statistics are known by both firms to be gaussian with mean $\bar{\alpha} > 0$ and variance $\xi > 0$, i.e. $\alpha \in N[\bar{\alpha},\xi]$. Each firm has access to information about the value of α through several noisy channels; and adopting an additive model, we define the observations of Firms 1 and 2 respectively by

$$y_1 = \alpha + w_3, \qquad w_3 \in N[0,s_3],\ s_3 > 0 \tag{12a}$$

$$y_2 = \alpha + w_4, \qquad w_4 \in N[0,s_4],\ s_4 > 0, \tag{12b}$$

where α, w_3 and w_4 are statistically independent, and $s_4 > s_3$.

Within the framework of the duopoly problem defined above, we again consider a single-stage nonzero-sum stochastic game, and further we assign zero cost functions to both firms (or we can assume identical linear cost functions for both firms -- linear in the production level of that firm -- the effect of which can always be absorbed by $\bar{\alpha}$). Firm i wants to maximize his average profit $\bar{P}_i$ under the assumptions that Firm j ($i \neq j$) acts rationally and that the process is noncooperative, that is, there can be no explicit collusion. The profit functions P_i are defined by

$$P_1(q,r) = qp \tag{13a}$$

$$P_2(q,r) = rp, \tag{13b}$$

and the average profit function $\bar{P}_i(q,r)$ for the i'th firm, for fixed $q(.)$, $r(.) \in \Sigma$ is given by

$$\bar{P}_i(q,r) = E\{P_i(q(y_1),r(y_2)) \,|\, q(.),r(.)\}, \; i=1,2, \tag{14}$$

where the expectation operation is taken over the prior statistics of α, w_3 and w_4.

The Nash (Cournot) equilibrium solution can be defined for this problem as in (4a) - (4b) but with the direction of the inequalities reversed, since the firms are now trying to maximize their respective average profits. Denoting the expected Nash profit of the i'th firm by $\bar{P}_i(q^*,r^*)$, it is obvious that $\bar{P}_1(q^*,r^*) = \bar{P}_2(q^*,r^*)$ for all β, $\bar{\alpha}$, $\xi > 0$ whenever $s_3 = s_4$.

However, when $s_3 < s_4$, we will show in the sequel that the Nash solution pair (q^*,r^*)has a property which completely contradicts with that of Problem 1:

Proposition 2:

The Nash (Cournot) solution pair for the duopoly problem of this section (Problem 2) is given for all $\bar{\alpha}$, β, ξ, s_3, $s_4 > 0$ by

$$q^*(y_1) = a_1^* y_1 + a_2^* \bar{\alpha} \tag{15a}$$

$$r^*(y_2) = b_1^* y_2 + b_2^* \bar{\alpha}\,, \tag{15b}$$

where a_1^*, a_2^*, b_1^* and b_2^* are given by

$$a_1^* = \xi(\xi+2s_4)/\{\beta[4(\xi+s_4)(\xi+s_3) - \xi^2]\}, \tag{16a}$$

$$a_2^* = \{s_3(\xi+s_4)(\tfrac{1}{\beta}+a_1^*) + s_4\xi(a_1^*-\tfrac{1}{\beta})\}/3(\xi+s_3)(\xi+s_4), \tag{16b}$$

$$b_1^* = \xi(\xi+2s_3)/\{\beta[4(\xi+s_4)(\xi+s_3) - \xi^2]\}, \tag{16c}$$

$$b_2^* = \{s_4(\xi+s_3)(\tfrac{1}{\beta}+b_1^*) + s_3\xi(b_1^*-\tfrac{1}{\beta})\}/3(\xi+s_3)(\xi+s_4). \tag{16d}$$

Proof:

Maximizing (14) over $q(.) \in \Sigma$, for i=1 and for fixed $r(.) \in \Sigma$ we obtain

$$q = \frac{1}{2\beta}\left[\frac{s_3}{\xi+s_3}\,\bar{\alpha} + \frac{\xi}{\xi+s_3}\,y_1\right] - \frac{1}{2}\,E\{r|y_1\}; \tag{17a}$$

and similarly, maximizing (14) over $r(.) \in \Sigma$, for i=2 and for fixed $q(.) \in \Sigma$ we get

$$r = \frac{1}{2\beta}\left[\frac{s_4}{\xi+s_4}\bar{\alpha} + \frac{\xi}{\xi+s_4} y_2\right] - \frac{1}{2} E\{q|y_2\}. \tag{17b}$$

We note parenthetically that equation (17a) is known as the reaction curve of Firm 1 when Firm 2 picks the policy $r(.) \varepsilon \Sigma$; a similar interpretation applies to (17b). The intersection of these two reaction curves will determine the Nash solution pair (q^*,r^*). Now, it follows from (17a) and (17b) that the Nash policies q^* and r^* will be linear in y_1 and y_2 respectively and also linear in $\bar{\alpha}$; hence letting $q = a_1y_1 + a_2\bar{\alpha}$, $r = b_1y_2 + b_2\bar{\alpha}$, we have

$$E\{r|y_1\} = \left(b_2+b_1\frac{s_3}{\xi+s_3}\right)\bar{\alpha} + b_1\frac{\xi}{\xi+s_3}y_1 \tag{18a}$$

$$E\{q|y_2\} = \left(a_2+a_1\frac{s_4}{\xi+s_4}\right)\bar{\alpha} + a_1\frac{\xi}{\xi+s_4}y_2. \tag{18b}$$

Substituting (18a) and (18b) into (17a) and (17b) respectively, we obtain the equations

$$a_1 = \frac{1}{2}\frac{\xi}{\xi+s_3}\left(\frac{1}{\beta}-b_1\right) \tag{19a}$$

$$a_2 = \frac{1}{2}\left[\frac{s_3}{\beta(\xi+s_3)} - b_2 - b_1\frac{s_3}{\xi+s_3}\right] \tag{19b}$$

$$b_1 = \frac{1}{2}\frac{\xi}{\xi+s_4}\left(\frac{1}{\beta} - a_1\right) \tag{19c}$$

$$b_2 = \frac{1}{2}\left[\frac{s_2}{\beta(\xi+s_4)} - a_2 - a_1\frac{s_4}{\xi+s_4}\right] \tag{19d}$$

and the solution to (19a) - (19d) is given by (16a) - (16d). Q.E.D.

Property 2:

The Nash equilibrium solution pair (15a) - (15b) for Problem 2 has the property that whenever $s_3 < s_4$, $\bar{P}_1(q^*,r^*) > \bar{P}_2(q^*,r^*)$ for all $\bar{\alpha} \geq 0$, $\beta>0$, $\xi > 0$.

Proof:

It follows from the definitions of $P_1(.,.)$ and $P_2(.,.)$ that

$$\bar{P}_1(q^*,r^*) - \bar{P}_2(q^*,r^*) = E\{(q^*-r^*)\alpha - (q^{*2}-r^{*2})\beta + (q^*-r^*)\bar{\alpha}\}. \tag{20a}$$

After substituting for q^*, r^* given by (15a) - (15b) in (20a), some straightforward but rather extensive manipulations yield:

$$\bar{P}_1(q^*,r^*) - \bar{P}_2(q^*,r^*) = \phi(s_4-s_3), \tag{20b}$$

where $\phi > 0$ is defined by

$$\phi = \xi^2/\beta[4(\xi+s_4)(\xi+s_3) - \xi^2]. \tag{20c}$$

Property 2 now follows from expression (20b), since $\phi > 0$ for all ξ, $\beta > 0$. Q.E.D.

Hence, roughly speaking, the Nash solution to the duopoly problem has the rather intuitive property that the firm who observes "better" has a higher profit in the market, which is completely contradictory with Property 1 of the solution of Problem 1.

For Problem 2, for the extreme case when Firm 1 makes perfect observation (i.e. $s_3 = 0$) and Firm 2 makes no observation (i.e. $s_4 = +\infty$), it follows from (20b) and (20c) that the average profit share of Firm 1 in the market is higher than that of Firm 2 by an amount $[\xi/4\beta]$.

IV. DISCUSSION OF THE TWO PROBLEMS

So far we have considered two stochastic game problems (Problems 1 and 2) the solutions to which have a contradictory property. In this section, we are going to imbed Problems 1 and 2 within the framework of a more general stochastic game problem (Problem 3) for the purpose of comparing the formulations as well as the properties of the solutions of these two problems. In what follows we are going to assume $\bar{\alpha} = 0$ in the formulation of Problem 2 of section III in order to conserve notation.

To this end, we consider a nonzero-sum stochastic game (Problem 3) defined by the quadratic cost functions

$$L_1(u,v) = (x+u+v)^2 c_1 + u^2 d_1 + v^2 d_2 + vxd_3 + x^2 c_2 \tag{21a}$$

$$L_2(u,v) = (x+u+v)^2 c_1 + v^2 d_1 + u^2 d_2 + uxd_3 + x^2 c_2, \tag{21b}$$

where c_1, c_2, d_1, d_2, d_3 are scalars known to both players, with $c_1 > 0$, $d_1 > 0$. x is a gaussian random vector with mean zero and variance $\sigma > 0$. Player 1 controls u through an observation $\tilde{z}_1$ and Player 2 controls v through an observation $\tilde{z}_2$, where

$$\tilde{z}_1 = hx + \tilde{w}_1, \qquad \tilde{w}_1 \in N[0,\tilde{s}_1], \qquad \tilde{s}_1 > 0 \tag{22a}$$

$$\tilde{z}_2 = hx + \tilde{w}_2, \qquad \tilde{w}_2 \in N[0,\tilde{s}_2], \qquad \tilde{s}_2 > 0 \quad , \tag{22b}$$

with x, $\tilde{w}_1$ and $\tilde{w}_2$ being statistically independent and $\tilde{s}_2 > \tilde{s}_1$.

Defining the average cost function $\bar{L}_i(u,v)$ for the i'th player in a similar way as in expression (3) for Problem 1, we seek the Nash solution pair (u^*,v^*) for Problem 3.

Problems 1 and 2 can now be thought of as special cases of this more general problem through the following identifications:

Identification 1: Problem 3 becomes identical to Problem 1 when $d_2 = d_3 = c_2 = 0$, $c_1 = c$, $d_1 = d$, $h = 1$, $\tilde{s}_1 \equiv s_1$, $\tilde{s}_2 \equiv s_2$.

Identification 2: Problem 3 becomes identical to Problem 2 when $c_1 = \frac{\beta}{2}$, $d_1 = \frac{\beta}{2}$, $x = -\frac{\alpha}{\beta}$, $d_2 = -\frac{\beta}{2}$, $d_3 = -\beta$, $c_2 = -\frac{\beta}{2}$, $h = -\beta$, $\sigma \equiv \xi$, $\tilde{s}_1 \equiv s_3$, $\tilde{s}_2 \equiv s_4$, $u \equiv q$, $v \equiv r$.

Proposition 3:

The Nash solution pair for Problem 3 of this section is given for all d_2, d_3, c_2, h and c_1, d_1, σ, $\tilde{s}_1$, $\tilde{s}_2 > 0$ by

$$u^*(\tilde{z}_1) = -\eta \frac{h\sigma}{h^2\sigma+\tilde{s}_1}\left(1 - \frac{c_1}{(c_1+d_1)} \cdot \frac{h^2\sigma}{(h^2\sigma+\tilde{s}_2)}\right)\tilde{z}_1 \tag{23a}$$

$$v^*(\tilde{z}_2) = -\eta \frac{h\sigma}{h^2\sigma+\tilde{s}_2}\left(1 - \frac{c_1}{(c_1+d_1)} \cdot \frac{h^2\sigma}{(h^2\sigma+\tilde{s}_1)}\right)\tilde{z}_2 \tag{23b}$$

with

$$\eta \triangleq \left(\frac{c_1}{c_1+d_1}\right)\Bigg/\left[1 - \left(\frac{c_1}{c_1+d_1}\right)^2 \cdot \frac{(h^2\sigma)^2}{(h^2\sigma+\tilde{s}_1)(h^2\sigma+\tilde{s}_2)}\right]. \tag{23c}$$

Proof:

The proof follows the same lines of thought employed in the proof of Proposition 1 if one notes that the reaction curves for Problem 3 are given by

$$u = -\frac{c_1}{c_1+d_1}\left[\frac{h\sigma}{h^2\sigma+\tilde{s}_1}\tilde{z}_1 + E\{v|\tilde{z}_1\}\right] \tag{24a}$$

$$v = -\frac{c_1}{c_1+d_1}\left[\frac{h\sigma}{h^2\sigma+\tilde{s}_2}\tilde{z}_2 + E\{u|\tilde{z}_2\}\right]. \tag{24b}$$

Q.E.D.

We parenthetically note at this point that using Identifications 1 and 2, the solution (23a) - (23b) simplifies down to solutions (5a) - (5b) and (15a) - (15b) respectively.

Property 3:

The Nash equilibrium solution pair (23a) - (23b) has the property that whenever $s_1 < s_2$,

$$\text{(i)}\ \bar{L}_1(u^*,v^*) > \bar{L}_2(u^*,v^*) \text{ if } \tilde{\kappa} > 0 \tag{25a}$$

$$\text{(ii)}\ \bar{L}_1(u^*,v^*) < \bar{L}_2(u^*,v^*) \text{ if } \tilde{\kappa} < 0, \tag{25b}$$

where

$$\tilde{\kappa} \triangleq \frac{c_1(d_1-d_2)}{c_1+d_1} + d_3. \tag{25c}$$

Proof:

Substituting the Nash policies (u^*,v^*) into the cost functions $L_1(u,v)$ and $L_2(u,v)$ and averaging over the statistics of the random variables involved, we obtain

$$\bar{L}_1(u^*,v^*) - \bar{L}_2(u^*,v^*) = \frac{\eta h^2\sigma^2\tilde{\kappa}(\tilde{s}_2-\tilde{s}_1)}{(h^2\sigma+\tilde{s}_1)(h^2\sigma+\tilde{s}_2)}$$

from which (25a) and (25b) follow, since η as defined by (23c) is a positive scalar. Q.E.D.

Now, using Identifications 1 and 2 it can be shown that $\tilde{\kappa}$ is always positive for Problem 1, whereas it is always negative for Problem 2. In fact, within the general framework of Problem 3, it is possible to consider two classes of game problems depending on the sign of $\tilde{\kappa}$, with $\tilde{s}_2 > \tilde{s}_1$. The first class is determined by (25a), and the problems belonging to that class have the property that the player who has a "better" observation ends up facing a *higher* average Nash cost; Problem 1 of section II belongs to this class. The second class is characterized by (25b), and the problems belonging to that class have the property that the player who makes "better" measurements ends up with a *lower* average Nash cost; Problem 2 of section III belongs to this class.

REFERENCES

[1] A. W. Starr and Y. C. Ho, "Nonzero-Sum Differential Games", *Journal of Optimization Theory and Applications*, Vol. 3, No. 3, (1969), 184-206.

[2] A. W. Starr and Y. C. Ho, "Further Properties of Nonzero-Sum Differential Games", *Journal of Optimization Theory and Applications*, Vol. 3, No. 4, (1969), 207-219.

[3] M. H. Foley and W. E. Schmitendorf, "On a Class of Nonzero-Sum, Linear-Quadratic Differential Games", *Journal of Optimization Theory and Applications*, Vol. 7, No. 5, (1971), 357-377.

[4] R. M. Cyert and M. H. DeGroot, "Multiperiod Decision Models with Alternating Choice as a Solution to the Duopoly Problem", *The Quarterly Journal of Economics*, Vol. LXXXIV, (August 1970), 410-429.

[5] R. M. Cyert and M. H. DeGroot, "Bayesian Analysis and Duopoly Theory", *The Journal of Political Economy*, Vol. 78, No. 5, (1970), 1168-1184.

HIERARCHICAL CONTROL OF OPTIMAL SUBSYSTEMS

P.D. SHEA, J.S. SHAFRAN, AND M.A. RADKE
TRW Systems
Redondo Beach, Calif.

ABSTRACT

An approximate feedback controller is developed for a large multilevel hierarchical system. This controller allocates assignments to optimal subsystems so that the system cost which is the sum of the subsystem costs are minimized.

Consider an operating system which is composed of a series of interconnected subsystems. Each subsystem has a bounded set of resources and a subsystem controller which allocates these resources among the assignments of the subsystem. It is assumed that each subsystem controller is constructed to minimize its cost function with constraints. The system or master controller has the freedom to transfer the assignments among the subsystems but it cannot directly control the resource allocation to the assignments. The system controller will attempt to minimize a system cost function which is a composite of the subsystem cost functions.

An example of this type of process is an antiballistic missile system. In this case the subsystem controller has the ability to allocate resources among the activities or assignments which are required to locate, identify, and intercept the attack. The subsystem performance index is the expected value of the loss due to unintercepted threatening reentry vehicles. The system controller has the ability to transfer the assignments among the units. The system performance index is the sum of the unit expected losses.

We will assume that the control problem is structured as a series of discrete parameter optimizations. Periodically the system controller will compute the optimum assignments and use these assignments until the next update. The subsystem controllers operate in a similar but much faster manner. The change in system assignments at each update will be assumed to be small.

In the case to be considered it is assumed that the cost is of the form

$$J = \sum_{n=1}^{N} \sum_{i=1}^{I_n} f_{ni}(u_{ni}, p_{ni}, \bar{x}) \tag{1}$$

where

u_{ni} = portion of assignment i allocated to unit n by the system controller
p_{ni} = portion of unit n resources allocated to assignment i
N = number of subsystems
I_n = number of assignments in unit N
$\bar{x}$ = state of the assignments
$f_{ni}(u_{ni}, p_{ni}, \bar{x})$ is convex and monotonic with respect to all u_{ni} and p_{ni}.

The subsystem level resource constraints can be expressed as

$$\sum_{i=1}^{I_n} p_{ni} = p_{nT} \quad \text{for each subsystem} \tag{2}$$

where p_{nT} is the amount of resource available.

The system controller constraint can be expressed as

$$\sum_{n=1}^{N} u_{ni} = u_{iT} \qquad \text{for each assignment } i \tag{3}$$

where u_{iT} is the total amount of assignment which must be allocated among the subsystems.

The problem could be formulated by appending the equality constraints to the cost function by Lagrange multipliers.[1]

$$L = \sum_{n=1}^{N} \sum_{i=1}^{I_n} f_{ni}(u_{ni},p_{ni},\bar{x}) + \sum_{i=1}^{I} \lambda_i (u_{nT} - \sum_{n=1}^{N} u_{ni}) + \sum_{n=1}^{N} \lambda_{sn}(p_{nT} - \sum_{i=1}^{I_n} p_{ni}) \tag{4}$$

For the type of cost function defined the necessary conditions for optimality are

$$\frac{\partial L}{\partial u_{ni}} = \frac{\partial f_{ni}}{\partial u_{ni}} - \lambda_i = 0 \tag{5}$$

$$\frac{\partial L}{\partial \lambda_i} = u_{nT} - \sum_{n=1}^{N} u_{ni} = 0 \tag{7}$$

$$\frac{\partial L}{\partial p_{ni}} = \frac{\partial f_{ni}}{\partial p_{ni}} - \lambda_{sn} = 0 \tag{6}$$

$$\frac{\partial L}{\partial \lambda_{sn}} = p_{nT} - \sum_{i=1}^{I_n} p_{ni} = 0 \tag{8}$$

for n=1 ... N and i=1 ... I where I is the number of assignments to be allocated.

Several important system considerations make this direct approach intractable. The cost function components f_{ni} are given in tabular form. This means the equations 5 through 8 must be solved by numerical techniques.

Another difficulty is the imposing dimensionality of the problem. Each subsystem could easily be processing hundreds of activities simultaneously and there could be many subsystems under the system controller. Under these conditions the solution of the necessary conditions at a high rate would be very time consuming. Another characteristic of the example is that the subsystem allocation must take place at a much faster rate than the system allocation. For these reasons we will require that the system controller must allocate resources without solving the subsystem allocation problem directly.[2]

We will assume that the subsystems solve equations 6 and 8 in an iterative manner by choosing values for λ_{sn}, finding the resulting p_{ni} from equation 6 and testing to see if equation 8 is satisfied.

Equation 5 can be expanded in terms available from the subsystems.

$$\frac{\partial L}{\partial u_{ni}} = \left.\frac{\partial f_{ni}}{\partial u_{ni}}\right|_{\lambda_{sn}} + \left.\frac{\partial f_{ni}}{\partial p_{ni}}\right|_{u_{ni}} \left.\frac{\partial p_{ni}}{\partial u_{ni}}\right|_{\lambda_{sn}} - \lambda_i = 0 \tag{9}$$

The first term on the right side of equation (9) will remain approximately constant for varying assignments. The constraint that is constant is equivalent to holding the Lagrange multiplier of the subsystem constant. The first part of the second term is the Lagrange multiplier of the subsystem. We know how this varies if p_{nT} is allowed to vary since this data was generated during subsystem iteration. The last part of the second term will be assumed to be a constant equal to the ratio between p_{ni} and u_{ni} at the previous subsystem allocation.

Using these approximations equation (9) can be rewritten as

$$\frac{\partial L}{\partial u_{ni}} = C_{ni} + \lambda'_{sn}(p'_{nT})\, D_{ni} - \lambda_i \cong 0 \tag{10}$$

$$\text{where } C_{ni} = \left.\frac{\partial f_{ni}}{\partial u_{ni}}\right|_{\lambda_{sn}}, \quad \lambda'_{sn}(p'_{nT}) = \left.\frac{\partial f_{ni}}{\partial p_{ni}}\right|_{u_{ni}}, \quad D_{ni} = \left.\frac{\partial P_{ni}}{\partial u_{ni}}\right|_{\lambda_{sn}}$$

p_{nT} amount of resource available at subsystem n.

We will further assume that the subsystems will allocate resources to changed assignments by the ratio $\frac{\partial p_{ni}}{\partial u_{ni}}$ and that this change is used to modify the constraint p_{nT}.

$$p'_{nT} \cong p_{nT} - \sum_{j=1}^{I_n} \frac{\partial p_{ni}}{\partial u_{nj}} \Delta u_{nj} \qquad \text{for } n=1 \ldots N \tag{11}$$

where Δu_{nj} is the change in assignments.

The constraint equation (7) can be rewritten as

$$\sum_{n=1}^{N} \Delta u_{ni} = 0 \qquad i=1 \ldots I_n \tag{12}$$

Equations 10, 11 and 12 can be solved to evaluate the change in assignments, Δu_{ni}, for the new allocation.

References

[1] M. Athans and P.L. Falb, Optimal Control, McGraw-Hill Book Co., New York, 1966.

[2] M.D. Mesarovic, D. Macko, and Y. Takahara, Theory of Hierarchial, Multilevel, Systems, Academic Press, New York, 1970.

ON A CLASS OF LINEAR THREE-PERSON COOPERATIVE DIFFERENTIAL GAMES -- POLLUTION CONTROL[†][‡]

JI-GUAN LIN
Department of Electrical Engineering and Computer Science
Columbia University
New York, New York 10027

Summary

The theory of N-person cooperative differential games can be applied to problems of societal interest; for example, pollution control. To illustrate, consider three factories which produce a certain kind of chemical for the market. Let y_i (z_i) denote the amount of product (pollutant) produced for a period T:

$$y_i = \int_0^T c_i x_i(t)dt, \quad (z_i = \int^T d_i x_i(t)dt) \quad i = 1, 2, 3. \tag{1}$$

where $x_i(t)$ represents the production state (with process control $u_i(t)$):

$$\dot{x}_i(t) = a_i x_i(t) + b_i u_i(t), \quad i = 1, 2, 3. \tag{2}$$

Let factory i be penalized by $r_i' z_i$ for producing pollutant z_i. Combined with production cost $r_i'' y_i$ and constant cost k_i, the total cost becomes

$$K_i(y_i) = r_i y_i + k_i \quad (r_i \overset{\Delta}{=} r_i' di/c_i + r_i'') \quad i = 1, 2, 3,$$

Suppose that production and sale of the chemical will be completely suspended, voluntarily or by court, if total production y exceeds Q, an amount corresponding to a level of pollution crisis. Suppose also that the behavior of the consumers is described by a decreasing unit-price function

$$\begin{aligned} p(y) &= h(Q - y), & 0 \le y \le Q; \\ &= 0, & y > Q. \end{aligned}$$

That is, consumers are aware of the hazard, and will boycott the product if total production reaches the crisis level. Subtracting $K_i(y_i)$ from

---

[†] This research was supported by the National Science Foundation under Grant NSF-GK-32701.

[‡] For the Tenth Annual Allerton Conference on Circuit and System Theory, University of Illinois; October 4-6, 1972.

$y_i p(y)$, we have as profits the following P_i's:

$$P_i(y_1, y_2, y_3) = -h\, y_i(y_1+y_2+y_3 - Q) - r_i y_i - k_i, \quad 0 \le y_1+y_2+y_3 \le Q,$$
$$= -r_i\, y_i - k_i\ , \quad y_1+y_2+y_3 \ge Q,$$
$$i = 1, 2, 3\ . \tag{3}$$

Suppose that factories therefore set q_i's as production limits:

$$0 \le y_i \le q_i\ , \qquad i = 1, 2, 3. \tag{4}$$

This is a 3-person differential game [1], with the factories as the players, the profits (3) as the payoffs, and the differential equation (2) and the inequalities (4) as the constraints. Cooperation between factories become essential when danger of enforced plant closing or negative profit exists in free competition. The worth function below can be used as a measure for the essentiality.

Like [1], I denotes the set of all N players. A coalition S is a subset S of I all players of which cooperate to maximize $\Sigma_S P_i(y_1, \ldots, y_N) \overset{\Delta}{=} P_S(y_1, \ldots, y_N)$. The worth w(S) of coalition S is defined as

$$w(S) = \max_{i \in S}\ \min_{j \notin S}\ P_S\,(\ldots,\ y_i\ \ldots,\ y_j\ \ldots)\ .$$

The worth w(S) is what coalition S will surely get at the worst case if they still cooperate to their best. The set function w(S), $S \subset I$ has been shown to be super-additive: $w(S) + w(T) \le w(S \cup T)$ for disjoint subsets S and T of I. Taking the singletons $\{i\}$, $i = 1, 2, \ldots, N$ as the disjoint subsets, we have then

$$\sum_{i=1}^{N} w(\{i\}) \le w(\{1, 2, \ldots, N\}) \overset{\Delta}{=} w(I)\ . \tag{5}$$

Now $w(\{i\})$ represents the payoff player i will receive when others are all against him but he still does his best. If he ever considers cooperation with others, he will require no less share of the collective outcome than $w(\{i\})$, then. Cooperation is thus inessential if equation in (5) holds and essential, otherwise.

For the above 3-person game, the worth function has been computed for $r_1 < r_2 < r_3$. The following observations are worth mentioning. If $2(q_1 + q_2 + q_3) + r_3/h \le Q$, or $Q \le r_1/h$, cooperation is inessential; in the former case, competition is free and full production up to q_i's is the optimal; in the latter, no production at all is the optimal. Between these extremes, whether or not cooperation is essential depends on the range of Q.

For example, cooperation is essential if

$$\max_{i=1,2,3} \{ q_i + r_i/h + (q_1+q_2+q_3) \} \le Q < r_3/h + 2(q_1+q_2+q_3),$$

or, $r_1/h < Q \le \min_{i=1,2,3} \{ r_i/h - q_i + (q_1+q_2+q_3) \}$.

The worth function w does not require the existence of values of games [3] as does the characteristic function v [1, 2]; it has been shown, however, that w is as effective a measure as v, not only for essentiality but also for domination and solutions. Fundamental-triangle method and other analysis in [1] can be applied equally well with v replaced by w. Non-inferiority and preferability are considered respectively for avoiding some technical difficulty of domination and for excluding undesirable solutions.

References

[1] Lin, J. G., "On N-Person Cooperative Differential Games," Proc. 6th Annual Princeton Conf. on Information Sciences and Systems, March 23-24, 1972, Electrical Engineering, Princeton Univ., pp. 502-507.

[2] Von Neumann, J. and Morgenstern, O., Theory of Games and Economic Behavior, 3rd ed. New York: Wiley, 1967.

[3] Lin, J. G. and Varaiya, P. P., "Existence of Saddle Points in Differential Games," SIAM J. Control, vol. 7, no. 1, 1969, pp. 141-157.

OPTIMALITY FOR A MULTICRITERION PERTURBED SYSTEM

A. HAURIE AND J.L. GOFFIN
Ecole des Hautes Etudes Commerciales
Montréal, Québec.

In our research, we study the problem of decision-making in engineering and economic systems subject to set constrained disturbances, and whose performance is described by a vector valued cost criterion.

The system is defined by :

(1) a closed decision set $X \subset E^n$
(2) a bounded perturbation set (P.S.) Θ
(3) a vector valued cost criterion :
$\psi : E^n \times \Theta \to E^p, (x,\theta) \to \psi(x,\theta) = (\psi_j(x,\theta))_{j=1,\dots,p}$.

1. DEFINITION OF OPTIMALITY

A decision $x^* \in X$ is said to be Sup-Pareto-Optimal (S.P.O.) if $\forall x \in X$,

$$\mathrm{Sup}\{\psi_j(x,\theta) : \theta\in\Theta\} \le \mathrm{Sup}\{\psi_j(x^*,\theta) : \theta\in\Theta\} \ \forall j = 1,\dots,p$$

implies

$$\mathrm{Sup}\{\psi_j(x,\theta) : \theta\in\Theta\} = \mathrm{Sup}\{\psi_j(x^*,\theta) : \theta\in\Theta\} \ \forall j = 1,\dots,p.$$

2. SCALARIZATION PROCESS

The main results in optimization under vector-valued criteria revolve around the process of scalarization [5,6]. In the presence of disturbances, the scalarization cannot, in general, be performed directly on the cost functions as for $\alpha_j > 0$, $j = 1,\dots,p$, the condition :

$$\mathrm{Sup}\{\Sigma\alpha_j\psi_j(x^*,\theta) : \theta\in\Theta\} \le \mathrm{Sup}\{\Sigma\alpha_j\psi_j(x,\theta) : \theta\in\Theta\} \ \forall x\in X$$

does not imply that x^* is S.P.O.. The scalarization must be applied to the modified cost functions :
$\varphi_j : E^n \to E, x \to \varphi_j(x) = \mathrm{Sup}\{\psi_j(x,\theta) : \theta\in\Theta\}$.

The results of Da Cunha and Polak [5], apply to the modified, unperturbed system with the vector cost criterion : $\varphi = (\varphi_j)_{j=1,\dots,p}$.

3. LOCAL CHARACTERIZATION OF S.P.O.

A characterization of S.P.O. which does not involve the computation of the modified cost φ but uses only the cost ψ and the P.S. Θ is desirable. We attempt to achieve this in the case of additive disturbances, I.E. $\Theta \subset E^n$ and the vector cost can be written as $\psi(x + \theta)$.

3.1. Linear Costs

Let $\{z_j : j=1,\dots,p\}$ be p given vectors in E^n ; then we define $\psi_j(x + \theta) = < z_j, x + \theta >$.

THEOREM 1 :

a) If X is convex and :
$C = \{x\in E^n : < z_j, x > \le 0, j=1,\dots,p\}$

is not a subspace, then if $x^* \in X$ is S.P.O. there exists a $z_o \in E^n$, and p scalars $\alpha_j \ge 0$, such that :

(1) $z_o = \Sigma\, \alpha_j z_j$
(2) $< z_o, x^* > = \mathrm{Min}\{< z_o, x > : x \in X\}$

b) If C is a subspace, every $x \in X$ is S.P.O..

THEOREM 2 :

Conversely, if there exists a $x^* \in X$, a $z_o \in E^n$, and p scalars $\alpha_j > o$, such that conditions (1) and (2) of theorem 1 are satisfied, then x^* is S.P.O..

3.2. Non linear case

Assume that each ψ_j is C^1 and convex, and that Θ is a bounded polyhedron.

THEOREM 3 :

Let $x^* \in X$, and suppose that for each j, there exists a unique $\theta_j \in \Theta$ such that :

$\mathrm{Max}\{\psi_j(x^* + \theta) : \theta \in \Theta\} = \psi_j(x^* + \theta_j)$;

Then we have :

a) If x^* is S.P.O., then there exists a vector $z_o \in E^n$ and p scalars $\alpha_j \geq o$ such that :

(1) $z_o = \Sigma\, \alpha_j \nabla_x \psi_j(x^* + \theta_j)$
(2) $< z_o, x^* > = \mathrm{Min}\{< z_o, x > : x \in X\}$.

b) If there exist a vector $z_o \in E^n$, and p scalars $\alpha_j > o$ such that (1) and (2) of a) hold, then x^* is S.P.O..

The aim of this research is the development of stronger necessary and sufficient conditions for the non linear case with more general perturbation sets. The results appear to depend strongly on the geometry of the perturbation set.

4. REFERENCES

[1] M.C. DELFOUR and A.K. MITTER, Reachability of Perturbed Linear Systems and Min Sup Problems, SIAM J. On Control, 7 (1969), 521-533.

[2] D.P. BERTSEKAS and I.B. RHODES, On the Minimax Reachability of Targets and Target Tubes, Automatica, 7 (1971), 233-247.

[3] J.D. GLOVER and F.C. SCHWEPPE, Control of Linear Dynamic Systems with Set Constrained Disturbances, IEEE Trans. on Control, AC-16 (1971), 411-423.

[4] L.A. ZADEH, Optimality and non scalar valued performance criteria, IEEE Trans-automatic control - Vol. AC8 pp. 59-60, Janvier 1063.

[5] N.O. da CUNHA and E. POLAK, Constrained minimization under vector-valued criteria in linear topological spaces, in "Mathematical theory of control". Academic Press (BALAKRISHNAN, NEUSTADT editors). (1967).

[6] J. PHILIP, Algorithms for the vector maximization problem, Mathematical Programming, 2 (1972), 207-229.

OPTIMAL CONTROL WITH INCOMPLETE STATE FEEDBACK

CLYDE F. MARTIN*
Ames Research Center, NASA
Moffett Field, California 94035

ABSTRACT

It is in general not possible to construct an optimal incomplete state feedback control which is independent of initial conditions.

RESULTS

One of the major limitations of the linear optimal control theory is that the entire state must be available for feedback. In practice this is seldom the case. Several authors have considered this problem by designing compensators or by using observors of one kind or another. In [1] the problem is attacked directly by minimizing a cost functional in which the control is assumed to be a linear function of only some of the states. This approach will be used here.

It will be assumed that the system is described by a linear autonomous controllable differential equation with a constant linear output:

L) $\dot{x}=Fx+Gu$, $x(0)=x_o$, and $y=Cx$. u will be assumed to be of the form $u(t)=R^{-1}G'Y(t)Cx(t)$ and the performance criteria will be, **P**) $J(u)=\int x'Qx+u'Ru$, where Q is non-negative definite and R is positive definite. The limits of integration will be from 0 to T unless otherwise noted.

Substituting for u in L) and **P**) we obtain L*) $\dot{x}=(F+GR^{-1}G'Y(t)C)x(t)$, $x(0)=x_o$ and **P***) $J(Y(t))=\int x'(t)(Q+C'Y(t)GR^{-1}G'Y(t)C)x(t)dt$. If **P***) is minimized with respect to $Y(t)$ the minimizing $Y(t)$ will be function of x_o. To remove the dependence on initial conditions let $\phi(t)$ be the fundamental matrix of L* and consider the matrix valued function

$$J^*(Y)=\int \phi'(t)[Q+C'Y'(t)GR^{-1}G'Y(t)C]\phi(t)dt. \qquad (1)$$

$J^*(Y)$ is symmetric and since the vector space of symmetric matrices is partially ordered by, $A\le B$ if and only if $B-A$ is non-negative definite, the theory of vector valued cost functionals is applicable. It will be shown in this paper that in general $J^*(\ )$ has no infimum and thus that for every Y there exist an initial condition x_o and Y_1 such that $x_o'J^*(Y)x_o > x_o'J^*(Y_1)x_o$.

Necessary and sufficient conditions for the existence of an optimal incomplete state feedback control which is independent of the initial conditions will now be derived.

The following theorem is a generalization of a LaGrange multiplier theorem and may be proved using the LaGrange multiplier theorem as found in [2]. Similar theorems may be found in [3] and the references there.
Theorem 1. Let f be continuously Frechet differentiable function from a Banach space B into a partially ordered finite dimensional vector space V with positive cone P. Assume that P is closed and has nonempty interior. Let $\mathbf{G}$ be a function from B into a Banach space Z. If J has a local infimum under the constaint $\mathbf{G}(x)=\theta$ at the regular point x_o then there exists a linear function $Z^*:Z\to V$ such that $J^\#(x_o)+Z^*\mathbf{G}^\#(x_o)=\theta$. "#" denotes the Frechet derivative.

Reformulating L*) and 1) so that Theorem 1 is applicable gives $J(X(t),Y(t))=\int X'(t)[Q+C'Y'(t)GR^{-1}G'Y(t)C]X(t)dt$ and $\mathbf{G}(X(t),Y(t))=X(t)-I-\int_0^t (F+GR^{-1}G'Y(s)C)X(s)ds$. Assume that the components of X and Y are continuous. Thus the domain of J is a Banach space and the range of J is the finite dimensional space of symmetric matrices. The range of $\mathbf{G}$ is the set of continuous matrix valued functions. It is a routine task to verify the regularity of $\mathbf{G}$.

---

*The author is an NRC postdoctoral research associate.

Theorem 1 assures the existence of a linear function z. For this case z can be shown to have the form $z(L(t))=\int dP(t)L(t)+L'(t)dP(t)$ where $P(t)$ is a symmetric matrix. Since A and B are independent and arbitrary the equation $J^{\#}(X_o,Y_o)(A,B)+z(\mathbf{G}^{\#}(X_o,Y_o)(A,B))=\theta$ yields the following two equations after suitable manipulations: $\dot{P}(t)+(F+GR^{-1}G'Y_o(t)C)P(t)+(Q+C'Y_o'(t)GR^{-1}G'Y_o(t)C)X_o(t)=0$ and $G'(Y_o(t)CX_o(t)+P(t))=0$.

Define $K(t)=P(t)X_o^{-1}(t)$. The preceding equations yield the following: $G'(Y_o(t)C+K(t))=0$ and $K(t)=F'K(t)+K(t)F-K(t)GR^{-1}G'K(t)+Q$. Thus it follows that if there is an optimal feedback law of the form $u(t)=R^{-1}G'Y_o(t)CX(t)$ then $G'Y_o(t)C=-G'K(t)$ where K is the solution of the Ricatti equation. Thus, in general, such an incomplete state feedback will fail to exist. Summarizing the preceding is the following theorem.

Theorem 2. A necessary and sufficient condition for 1) to have an infimum $Y_o(t)$ is that $u(t)=R^{-1}G'Y_o(t)CS(t)$ minimizes $\mathbf{P}$).

In [1] the assumption was made that $Y_o(t)$ is independent of any particular initial value but dependent on the distribution of the initial conditions which were assumed to be uniformly distributed on the surface of a unit sphere. This led to an attempt to minimize the functional $J(Y(t))=\mathrm{tr}\int\phi'(t)[Q+C'Y(t)GR^{-1}G'Y(t)C]\phi(t)dt$.

Assume that the initial conditions are contained in a set Ω and have distribution function f. The expected value of

$$J(Y)=\int x_o'\phi'(t)[Q+C'Y'(t)GR^{-1}G'Y(t)C]\phi(t)x_o dt \tag{2}$$

is given by $E_{x(o)}x_o'W(Y)x_o=\int x_o'W(Y)x_o f(x_o)dx_o$ where $W(Y)$ is given by (1). Thus $E_{x(o)}x_o'W(Y)x_o$ is a positive linear functional on the space of symmetric matrices and there exists a finite set of x_i's such that $E_{x(o)}x_o'W(Y)x_o=\Sigma x_i'W(Y)x_i$. Minimizing the expected value of (2) is then equivalent to minimizing

$$J(Y)=\Sigma x_i'\int\phi'(t)[Q+C'Y'(t)GR^{-1}G'Y(t)C]\phi(t)dt\ x_i. \tag{3}$$

It is interesting to note that minimizing all functions of the form of (3) is equivalent to obtaining the non-inferior points of (1). At present the question of the existence of a minimizing Y for (3) has not been solved. In [1] a partial answer for (2) is given. The problem of obtaining a solution of (3) appears to be difficult even if it exists.

BIBLIOGRAPHY

1. Levine, William S., Timothy L. Johnson, Michael Athans, Optimal Limited State Variable Feedback Controllers for Linear Systems, IEEE Transactions on Automatic Control, 16, 1971, p. 785-793.
2. Luenberger, David G., Optimization by Vector Space Methods, 1969, John Wiley and Sons, Inc., New York.
3. Geering, H. P. and M. Athans, Necessary and Sufficient Conditions for Differentiable Non-Scalar-Valued Functions to Attain Extrema, JACC, Stanford University, Stanford, California, 1972.

INDEX OF AUTHORS

Ahmed, N. U. 532
Anderson, R. H. 2
Apfelbaum, M. J. 404
Averitt, S. F. 608

Bach, R. E., Jr. 507
Bahl, L. 185
Bajaj, K. S. 676
Baliga, B. V. 76
Banda, J. M. 335
Barman, J. F. 205
Basar, T. 819
Beckhoff, G. F. 471
Beltrami, E. 143
Berger, J. 792
Bodin, L. 143
Bourquin, J. J. 667
Bredeson, J. G. 637
Brown, D. P. 692
Burghart, J. H. 520

Callier, F. M. 205
Cannon, J. R. 769
Caron, J. W. 357
Caudill, P. J. 283
Chan, W. L. 498
Chandy, M. 534
Chang, C. L. 11, 20
Chang, H. Y. 752
Chang, S. K. 277
Chen, Y. E. 554
Chen, Z. 248
Cheung, L. S. 554
Chien, R. T. 175
Chow, S. G. 231
Chuang, Y. H. 133, 647
Chyung, D. H. 530
Cocke, J. 185
Costello, D. 178

Das, S. 647
Davison, E. J. 196
De Mori, R. 616
De Santis, R. M. 221
DePalma, G. F. 742
Denning, P. J. 724
Derin, H. 293
Desoer, C. A. 205
Dietrich, G. W. 742
DuCasse, E. G. 625, 635
Dwyer, S. J., III 283

Eastwood, L. F., Jr. 315
Ewing, D. J., Jr. 498

Forney, C. David, Jr. 186
Foudriat, E. C. 95
Frye, C. B., Jr. 404
Fu, K. S. 248, 257

Gault, J. W. 608
Goffin, J. L. 834
Grasselli, O. M. 211
Grosky, W. I. 452
Guidorzi, R. P. 39
Gulcur, H. O. 579
Gupta, N. H. 367

Hammond, W. M. 596
Harger, R. O. 97, 313
Harlow, C. A. 283
Harmon, L. D. 1
Hartmann, C. 166
Haurie, A. 834
Hayes, R. A. 667
Houdeshell, J. 367
Hulina, P. T. 637
Hsu, H. T. 416

Jamshidi, M. 56
Jankauskas, L. E. 568
Jelinek, F. 178, 185

Kalata, P. 99
Kaliski, M. E. 282
Kamal, S. 111
Kashdan, N. 282
Kelly, W. J. 481
Kim, M. C. 313
Klein, R. E. 784
Kleitman, D. J. 152
Ko, W. H. 791
Koivo, H. N. 231
Krause, K. L. 545
Kundu, S. 562
Kurimoto, K. 97
Kurth, C. F. 810

Leake, R. J. 481
Lee, R. C. T. 11
Lee, S. C. 649
Lee, T. N. 692
Levine, W. S. 486
Lin, J. G. 238, 831
Lodaya, M. D. 498
Loew, M. H. 257
Long, J. L. 659
Ludeman, L. C. 66
Lugannani, R. 315, 568

McAffe, L. C., Jr. 383
McLaren, R. W. 543

Marro, G. 39
Martin, C. F. 836
Mayeda, W. 159

Meditch, J. S. 68
Mehrotra, K. G. 443
Mersereau, R. M. 326
Merwin, R. E. 733
Metz, L. D. 774
Metze, G. A. 625, 635
Meyer, A. U. 579
Michalopoulos, D. A. 269
Mickle, M. H. 344, 351
Miller, W. C. 209
Mitra, D. 709
Mokari-Bolhassan, M. E. 392
Monopoli, R. V. 496
Montgomery, H. D. 709
Murata, T. 685

Nayak, R. P. 95
Nelson, C. T. 802
Nestor, R. G. 283

Opacic, J. 414

Page, C. V. 111
Paphitis, G. A. 685
Pastrick, H. L. 85
Post, S. E. 507
Priemer, R. 99

Radke, M. A. 828
Ramamoorthy, C. V. 759
Ransom, M. N. 48
Rao, T. R. N. 426
Raviv, J. 185
Reddi, S. S. 759
Reddy, S. M. 121, 129
Roy, T. S. 357
Rubin, I. 303
Rudolph, L. 166, 443
Rychlewski, W. J. 543

Saeks, R. 48
Sandler, M. 587
Sato, Y. 598
Scarborough, S. M. 403
Schaffner, M. R. 462
Schwarzlander, H. 443
Shafran, J. S. 828
Shah, K. R. 76
Shea, P. D. 828
Shekel, J. 700
Shen, V. Y. 545
Sid-Ahmed, M. 209
Sinha, N. K. 509
Stewart, T. L. 596
Sypherd, A. D. 650

Tabak, D. 357, 414
Thomas, J. B. 293
Trick, T. N. 659
Tsai, S. 434
Tucker, J. E. 507

Van Slyke, R. 165
Varanasi, M. R. 426
Vishnubhotla, S. R. 133
Vogt, W. G. 344, 351

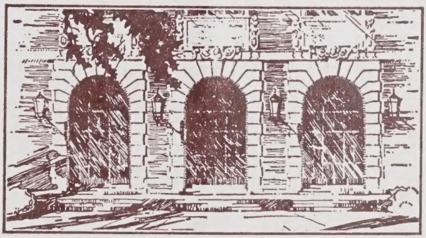
Wan, L. A. 374
Wan, S. J. 374
Wang, S. H. 196
Weissenberger, S. 29
Welch, R. V. 486
Wong, C. K. 271, 277
Wozny, M. J. 783

Yang, S. C. 101
Yau, S. S. 101, 742
Yavin, Y. 76
Yue, P. C. 271

DEPARTMENT OF
COMPUTER SCIENCE
LIBRARY

LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

NOTICE: According to Sec. 19 (a) of the University *Statutes*, all books and other library materials acquired in any manner by the University belong to the University Library. When this item is no longer needed by the department, it should be returned to the Acquisition Department, University Library.



UNIVERSITY OF ILLINOIS-URBANA
621.31921AL54P C003
PROCEEDINGS.$URBANA ILL
10 1972
3 0112 006279563